

KLASIFIKASI JUDUL TUGAS AKHIR MENGGUNAKAN TEKNOLOGI WEB SEMANTIK



الجامعة الإسلامية
الاندونيسية

Disusun Oleh:

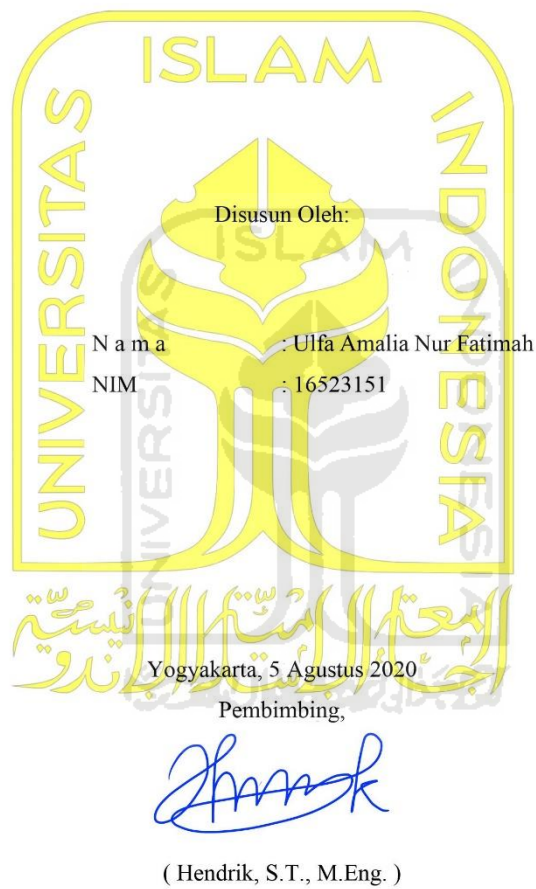
N a m a : Ulfa Amalia Nur Fatimah
NIM : 16523151

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2020**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**KLASIFIKASI JUDUL TUGAS AKHIR MENGGUNAKAN
TEKNOLOGI WEB SEMANTIK**

TUGAS AKHIR



Disusun Oleh:

Nama : Ulfa Amalia Nur Fatimah
NIM : 16523151

Yogyakarta, 5 Agustus 2020

Pembimbing,

Hendrik

(Hendrik, S.T., M.Eng.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**KLASIFIKASI JUDUL TUGAS AKHIR MENGGUNAKAN
TEKNOLOGI WEB SEMANTIK**

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk
memperoleh gelar Sarjana Komputer dari Program Studi Informatika
di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 5 Agustus 2020

Tim Penguji

Hendrik, S.T., M.Eng.

Anggota 1

Dr. Raden Teduh Dirgahayu, S.T., M.Sc.

Anggota 2

Irving V. Papatungan, S. T., M.Sc., Ph.D.


 Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Ulfa Amalia Nur Fatimah
NIM : 16523151

Tugas akhir dengan judul:

**KLASIFIKASI JUDUL TUGAS AKHIR MENGGUNAKAN
TEKNOLOGI WEB SEMANTIK**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 5 Agustus 2020



(Ulfa Amalia Nur Fatimah)

HALAMAN PERSEMBAHAN

Dengan menyebut nama Allah, Maha Pengasih, Maha Penyayang, lagi Maha Cinta.

Puji syukur bagi Allah karena dengan nikmat-Nya dan kuasa-Nya telah mengizinkan saya melewati salah satu fase penting dalam hidup saya.

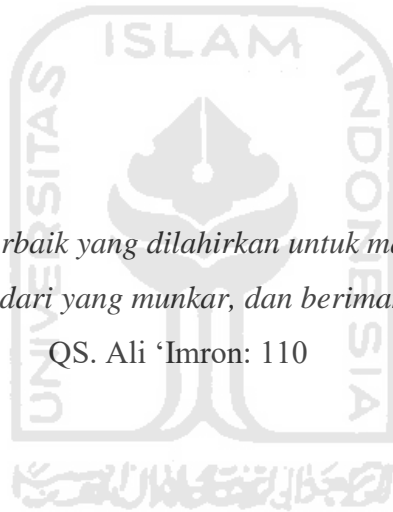


Saya persembahkan karya ini untuk kedua orangtua saya, kakak-kakak, serta semua pihak yang mendukung dan membantu saya. Semoga Allah membalas semua kebaikan kalian. *Aamiin yaa rabbal 'alamiin.*

HALAMAN MOTO

“Kamu adalah umat yang terbaik yang dilahirkan untuk manusia, menyuruh kepada yang ma'ruf, mencegah dari yang munkar, dan beriman kepada Allah.”

QS. Ali 'Imron: 110



KATA PENGANTAR

Assalamualaikum warahmatullahi wabarakaatuh.

Alhamdulillah rabbil 'alamiin. Segala puji bagi Allah *subhanahu wa ta'ala* atas segala nikmat, karunia, serta hidayah-Nya sehingga tugas akhir berjudul **“Klasifikasi Judul Tugas Akhir Menggunakan Teknologi Web Semantik”** ini dapat diselesaikan sebagaimana mestinya. Tugas akhir ini dibuat sebagai salah satu syarat yang harus dipenuhi untuk menyelesaikan pendidikan jenjang sarjana dari Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Dalam proses penyelesaian tugas akhir ini, tidak sedikit rintangan dan hambatan yang dihadapi, baik dari diri penulis sendiri maupun dari faktor lain. Namun, berbagai rintangan dan hambatan tersebut akhirnya dapat dilalui dikarenakan dukungan dari berbagai pihak. Maka dari itu, penulis ingin mengucapkan terimakasih kepada:

1. Kedua orangtua tercinta saya, Murhayanto dan Fadilah, yang senantiasa memberikan dukungan dan doa yang sangat berarti untuk saya.
2. Keempat kakak saya, Dedy, Shidiq, Ari, dan Dina yang selalu memotivasi saya.
3. Bapak Prof. Fathul Wahid, S.T., M.Sc., Ph.D., selaku Rektor Universitas Islam Indonesia.
4. Bapak Prof. Dr. Ir. Hari Purnomo, M.T., selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
5. Bapak Hendrik, S.T., M.Eng., selaku Ketua Jurusan Informatika Universitas Islam Indonesia sekaligus dosen pembimbing yang telah berkenan membimbing dan memberikan saran serta masukan untuk penulis.
6. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Program Studi Informatika – Program Sarjana, Universitas Islam Indonesia.
7. Ibu Novi Setiani, selaku dosen pembimbing akademik yang telah memberikan ilmu, arahan, serta saran kepada penulis.
8. Sahabat-sahabat saya selama kuliah; Fionna Saphira, Assyifa Narulita, Sabika Amalina, Sabila Amalia, Sari Kurnia, Puspita Dewi, Khoirunnisa Shofwatul, dan Rizal Hamdan yang senantiasa memberi semangat, memotivasi, menjadi tempat bercerita, berdiskusi, dan bersedia membantu dalam pengerjaan tugas akhir ini.
9. Kakak-kakak tingkat yang baik; Kartika Rizqi dan Sri Yuniarti yang senantiasa membantu saya dan memotivasi saya untuk menyelesaikan penelitian ini.

10. Ketiga belas teman-teman spesial saya, yang selalu memberikan saya senyum, warna, dan semangat dalam kehidupan saya.
11. Seluruh teman Hexadecima yang selalu mendukung dan memberi semangat. Terimakasih atas waktu-waktu berharganya.
12. Pihak-pihak lain yang telah membantu kelancaran proses penelitian ini yang tidak dapat disebutkan satu persatu.

Dengan selesainya penulisan laporan tugas akhir ini, besar harapan penulis agar karya ini dapat memberikan manfaat dan dapat dikembangkan untuk menjadi lebih baik lagi di masa mendatang.

Wassalamualaikum wa rahmatullahi wabarakaatuh.

Yogyakarta, 5 Agustus 2020



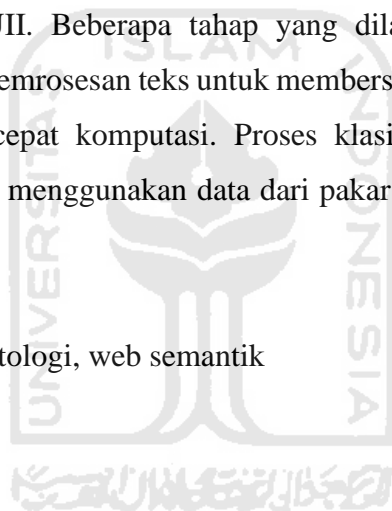
A handwritten signature in black ink, appearing to be 'Ulfa Amalia Nur Fatimah'.

(Ulfa Amalia Nur Fatimah)

SARI

Klasifikasi teks merupakan salah satu penelitian yang banyak dilakukan, baik untuk sektor pendidikan, maupun sektor lainnya. Sektor pendidikan, contohnya, dapat diterapkan untuk mengklasifikasikan teks judul tugas akhir mahasiswa untuk menggolongkannya ke beberapa konsentrasi yang telah ditentukan. Tugas tersebut dapat mempermudah dosen untuk memeriksa apakah proposal tugas akhir mahasiswa sudah sesuai dengan konsentrasi yang dipilihnya atau belum. Peneliti menggunakan pendekatan teknologi web semantik, yaitu ontologi, untuk proses klasifikasi. Ontologi yang dikembangkan bersumber dari *Association Computer Machinery* (ACM) dan *Computer Science Ontology* (CSO). Ontologi lain yang menjadi rujukan diantaranya *Medical Subject Headings* (MeSH), *National Cancer Institute*, dan *International Classification of Disease Version 10* (ICD10) yang disesuaikan dengan Panduan Kurikulum 2010 yang ada di Informatika UII. Beberapa tahap yang dilakukan sebelum masuk ke klasifikasi diantaranya adalah pra pemrosesan teks untuk membersihkan kata-kata yang kurang penting sekaligus untuk mempercepat komputasi. Proses klasifikasi memanfaatkan kueri SPARQL. Pengujian penelitian ini menggunakan data dari pakar yang menghasilkan akurasi sebesar 61,53%.

Kata kunci: klasifikasi teks, ontologi, web semantik



GLOSARIUM

| | |
|--------------|---|
| Frasa | gabungan terdiri dari dua kata atau lebih yang ukurannya lebih besar dari kata dan lebih kecil dari kalimat. |
| Klasifikasi | menetapkan sebuah objek ke dalam suatu kelas yang telah didefinisikan sebelumnya. |
| Kueri | sekumpulan instruksi untuk mengekstraksi atau mengambil data dari pangkalan data. |
| Literal | digunakan untuk nilai-nilai, seperti <i>string</i> , angka, dan tanggal. |
| Ontologi | model yang menggambarkan konsep dari domain tertentu secara eksplisit. |
| OWL | singkatan dari <i>Web Ontology Language</i> , yaitu bahasa web semantik yang dirancang untuk mewakili pengetahuan yang kaya dan kompleks tentang hal-hal, kelompok sesuatu, dan hubungan antara hal-hal tersebut. |
| Permutasi | penyusunan kembali secara spesifik objek-objek yang berbeda dari urutan sebenarnya. |
| RDF | singkatan dari <i>Resource Development Framework</i> , yaitu format yang digunakan untuk menyimpan data di web semantik. |
| Semantik | studi linguistik dan filosofis tentang makna dalam bahasa, pemrograman, logika formal, dan semiotik. |
| SPARQL | singkatan dari <i>SPARQL Protocol and The RDF Query Language</i> , yaitu bahasa kueri semantik yang dapat mengambil dan memanipulasi data yang tersimpan pada format RDF. |
| URI | singkatan dari <i>Uniform Resource Identifier</i> , yaitu serangkaian karakter yang digunakan untuk mengidentifikasi sumber dari nama, lokasi, atau keduanya. |
| Web Semantik | web yang memanfaatkan kemampuan semantik yang dimana mampu menemukan, menghubungkan, dan menggabungkan informasi di internet. |

DAFTAR ISI

| | |
|---|-----------|
| HALAMAN JUDUL | i |
| HALAMAN PENGESAHAN DOSEN PEMBIMBING | ii |
| HALAMAN PENGESAHAN DOSEN PENGUJI | iii |
| HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR..... | iv |
| HALAMAN PERSEMBAHAN | v |
| HALAMAN MOTO | vi |
| KATA PENGANTAR..... | vii |
| SARI | ix |
| GLOSARIUM | x |
| DAFTAR ISI | xi |
| DAFTAR TABEL | xiii |
| DAFTAR GAMBAR..... | xiv |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Pertanyaan Penelitian | 3 |
| 1.3 Batasan Masalah | 3 |
| 1.4 Tujuan Penelitian | 3 |
| 1.5 Manfaat Penelitian | 4 |
| 1.6 Metode Penelitian..... | 4 |
| 1.7 Sistematika Penulisan | 5 |
| BAB II LANDASAN TEORI | 6 |
| 2.1 Tinjauan Pustaka | 6 |
| 2.2 Landasan Teori..... | 11 |
| 2.2.1 Bahasa | 11 |
| 2.2.2 NLP | 13 |
| 2.2.3 Permutasi | 15 |
| 2.2.4 Klasifikasi..... | 15 |
| 2.2.5 Web Semantik | 16 |
| 2.2.6 Ontologi..... | 27 |
| BAB III METODOLOGI PENELITIAN | 33 |
| 3.1 Langkah-langkah Penelitian..... | 33 |
| 3.2 Deskripsi Langkah-langkah Penelitian | 34 |
| 3.2.1 Kajian Pustaka..... | 34 |
| 3.2.2 Pengembangan Ontologi | 34 |
| 3.3 Tahapan Klasifikasi..... | 37 |
| 3.3.1 Pra pemrosesan Teks (<i>Text Preprocessing</i>) | 38 |
| 3.3.2 Penemuan Kombinasi Kata (Frasa) Menggunakan Permutasi | 39 |
| 3.4 Pengujian..... | 40 |
| BAB IV HASIL DAN PEMBAHASAN..... | 42 |
| 4.1 Implementasi Ontologi..... | 42 |
| 4.2 Implementasi Klasifikasi..... | 47 |
| 4.2.1 Pra pemrosesan Teks Judul | 48 |
| 4.2.2 Permutasi | 49 |
| 4.2.3 Implementasi SPARQL..... | 50 |
| 4.3 Pengujian..... | 59 |
| BAB V KESIMPULAN DAN SARAN | 62 |
| 5.1 Kesimpulan | 62 |

5.2 Saran.....62
DAFTAR PUSTAKA.....63
LAMPIRAN66



DAFTAR TABEL

| | |
|---|----|
| Tabel 2. 1 Tinjauan Pustaka..... | 7 |
| Tabel 3. 1 Daftar Konsentrasi pada Informatika UII..... | 36 |
| Tabel 4. 1 Hasil pengujian menggunakan <i>Multiple Class Confusion Matrix</i> (1) | 59 |
| Tabel 4. 2 Hasil pengujian menggunakan <i>Multiple Class Confusion Matrix</i> (2) | 60 |



DAFTAR GAMBAR

| | |
|--|----|
| Gambar 1. 1 Diagram Hasil Kuisisioner | 2 |
| Gambar 2. 1 Hierarki Bahasa..... | 11 |
| Gambar 2. 2 Proses Pra Pemrosesan Teks | 14 |
| Gambar 2. 3 Teknologi Web Semantik..... | 17 |
| Gambar 2. 4 Ilustrasi Elemen Kompleks XML | 18 |
| Gambar 2. 5 Penulisan XML | 18 |
| Gambar 2. 6 Penggunaan Atribut di dalam Elemen XML..... | 19 |
| Gambar 2. 7 Contoh <i>Namespace</i> dalam XML..... | 19 |
| Gambar 2. 8 URI, URL, dan URN dalam Satu Kesatuan..... | 20 |
| Gambar 2. 9 Contoh RDF <i>Graph</i> yang Merepresentasikan <i>Triple</i> | 21 |
| Gambar 2. 10 RDF dalam format XML..... | 22 |
| Gambar 2. 11 Contoh OWL pada Dokumen RDF/XML..... | 24 |
| Gambar 2. 12 Deskripsi RDFS pada properti <code>owl:complementOf</code> | 25 |
| Gambar 2. 13 Deskripsi RDFS pada <i>instance</i> <code>Human</code> dan <code>FatherOfGirls</code> | 25 |
| Gambar 2. 14 Implementasi SPARQL..... | 26 |
| Gambar 2. 15 Hasil kueri SPARQL..... | 27 |
| Gambar 2. 16 <i>Instances</i> | 28 |
| Gambar 2. 17 Properti..... | 29 |
| Gambar 2. 18 <i>Functional Properties</i> | 29 |
| Gambar 2. 19 <i>Inverse Functional Properties</i> | 30 |
| Gambar 2. 20 <i>Transitive Properties</i> | 30 |
| Gambar 2. 21 <i>Symmetric Properties</i> | 31 |
| Gambar 2. 22 <i>Asymmetric Properties</i> | 31 |
| Gambar 2. 23 <i>Reflexive Properties</i> | 32 |
| Gambar 2. 24 <i>Irreflexive Properties</i> | 32 |
| Gambar 3. 1 Bagan Langkah-langkah Penelitian | 33 |
| Gambar 3. 2 Gambar Tampilan Website <i>Association for Computing Machinery</i> (ACM) | 35 |
| Gambar 3. 3 Taksonomi dari Domain <i>Computer Science</i> | 36 |
| Gambar 3. 4 Penerapan <i>subClassOf</i> pada Ontologi..... | 37 |
| Gambar 3. 5 Contoh Penerapan Pra Pemrosesan Teks | 38 |
| Gambar 3. 6 Contoh <i>Multiple Class Confusion Matrix</i> | 41 |

| | |
|---|----|
| Gambar 4. 1 Logo Protege | 42 |
| Gambar 4. 2 Tampilan awal Protege..... | 42 |
| Gambar 4. 3 Membuat subkelas dari <i>Superclass Thing</i> | 43 |
| Gambar 4. 4 Contoh subkelas baru pada InformationSystem | 43 |
| Gambar 4. 5 Subkelas versi Indonesia | 44 |
| Gambar 4. 6 Deskripsi <i>Equivalent To</i> pada subkelas InformationSystem terhadap SistemInformasi | 44 |
| Gambar 4. 7 <i>Reasoner</i> pada Protege..... | 45 |
| Gambar 4. 8 Ontologi yang setelah di- <i>reasoning</i> | 45 |
| Gambar 4. 9 Pemberian label pada subkelas | 46 |
| Gambar 4. 10 Format penyimpanan ontologi dalam Protege | 46 |
| Gambar 4. 11 Format penyimpanan dalam RDF/XML | 46 |
| Gambar 4. 12 Format penyimpanan dalam <i>Turtle Syntax</i> | 46 |
| Gambar 4. 13 Visualisasi ontologi..... | 47 |
| Gambar 4. 14 Kode program pra pemrosesan teks judul dengan bahasa pemrograman Java .48 | |
| Gambar 4. 15 Kode program proses permutasi dengan bahasa pemrograman Java..... | 49 |
| Gambar 4. 16 Logo Apache Jena | 50 |
| Gambar 4. 17 Logo Apache Lucene | 50 |
| Gambar 4. 18 Kode program proses <i>reasoning</i> pada ontologi | 51 |
| Gambar 4. 19 Kode program kueri SPARQL..... | 52 |
| Gambar 4. 20 Kode program eksekusi kueri SPARQL | 53 |
| Gambar 4. 21 Kode program untuk menampilkan kesimpulan akhir | 54 |
| Gambar 4. 22 Hasil akhir keluaran eksekusi kueri SPARQL setelah diurutkan berdasarkan nilai <code>countKonsentrasi</code> | 54 |
| Gambar 4. 23 Tampilan <i>search bar</i> pada GUI Program Pencarian Konsentrasi..... | 54 |
| Gambar 4. 24 Sebagian tampilan GUI pada hasil Program Pencarian Konsentrasi | 55 |
| Gambar 4. 25 Gambaran struktur ontologi subkelas dan kelas pada kueri SPARQL | 57 |
| Gambar 4. 26 Salah satu hasil <i>break down</i> dari kueri SPARQL pada frasa “sistem informasi” | 58 |

BAB I PENDAHULUAN

1.1 Latar Belakang

Istilah “Revolusi Industri 4.0” merupakan istilah yang sering kita dengar akhir-akhir ini. Secara tidak sadar, pemanfaatan teknologi, lambat laun, mendominasi segala aspek kehidupan manusia. Dengan kemampuan teknologi informasi yang cepat dan mudah diakses ini manusia dapat memperoleh segala informasi dan layanan yang diinginkan. Salah satu contoh pemanfaatan teknologi informasi yang populer adalah pemanfaatan portal web.

Portal web merupakan situs web yang dirancang khusus sebagai titik akses tunggal untuk pencarian informasi. Kita dapat melakukan pencarian, pengelolaan, pengintegrasian informasi, dan lainnya. Portal web mengumpulkan banyak informasi dari sekian banyak sumber informasi lalu dikumpulkan dalam satu tempat. Di ruang lingkup organisasi, portal web dapat disesuaikan dengan peran (*role*) dari pengguna yang didapatkan (Rosario et al., 2016).

Di sektor pendidikan, khususnya, penggunaan sebuah portal web untuk mengelola dokumen sudah banyak dilakukan di perguruan tinggi-perguruan tinggi di dunia, termasuk di Indonesia. Portal web tersebut memfasilitasi mahasiswa dan dosen untuk mempermudah mengelola dokumen secara elektronik, termasuk mengklasifikasikan atau mengelompokkan dokumen-dokumen sesuai dengan jurusan maupun konsentrasi penelitian. Di samping itu, pencarian dokumen lebih mudah dilakukan dengan menggunakan portal web.

Program Studi (Prodi) Informatika Fakultas Teknologi Industri Universitas Islam Indonesia (Informatika FTI UII), khususnya Program Sarjana, adalah salah satu yang menerapkan penggunaan portal web untuk mengelola dokumen, khususnya untuk tugas akhir mahasiswa. Nama portal tersebut adalah Kerja Praktik dan Tugas Akhir Fakultas Teknologi Industri Universitas Islam Indonesia (KPTA FTI UII). Proposal yang diajukan tersebut diunggah sebagai persyaratan tugas akhir mahasiswa. Penggunaan portal ini dinilai cukup membantu dan mudah digunakan bagi mahasiswa dan dosen pembimbing. Mahasiswa yang sudah memiliki akun KPTA FTI UII dapat masuk dan mengunggah proposalnya, kemudian diseleksi oleh para dosen pengampu konsentrasi. Namun, yang menjadi masalah adalah sering kali ditemukan banyak mahasiswa yang keliru memilih konsentrasi untuk proposal tugas akhirnya sehingga dosen perlu mengecek apakah proposal tersebut sesuai dengan konsentrasi yang dipilihnya atau tidak. Jika belum sesuai dengan konsentrasi yang dipilih, kemungkinan

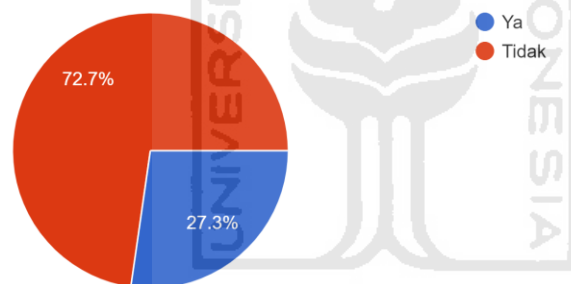
proposal tersebut perlu direvisi dan dapat menghambat pengerjaan skripsi mahasiswa. Dilihat dari permasalahan tersebut, maka perlu adanya model untuk proses *review* agar dapat mempermudah dosen memeriksa proposal tugas akhir secara teliti.

Proses identifikasi masalah menggunakan kuesioner yang disebarakan pada mahasiswa Informatika UII dengan *range* angkatan 2012 hingga 2014 mengenai portal KPTA FTI UII yang mereka gunakan. Penulis mengambil sampel dari ketiga angkatan tersebut dengan detail tujuh orang dari angkatan 2012, tiga orang dari angkatan 2013, dan satu orang dari angkatan 2014.

Salah satu pertanyaan kuesioner tersebut adalah apakah responden mengalami kesulitan dalam menentukan konsentrasi yang sesuai dengan isi proposal yang diajukan. Hasil dari pertanyaan tersebut ditunjukkan dengan Gambar 1. 1.

Apakah Anda pernah mengalami kesulitan dalam menentukan konsentrasi yang sesuai dengan proposal yang Anda ajukan?

11 responses



Gambar 1. 1 Diagram Hasil Kuisisioner

Hasil jawaban YA adalah sebanyak 27.3% atau tiga orang. Dari tiga orang tersebut mengutarakan alasan kesulitannya, yaitu dua dari tiga orang tersebut mengaku kesulitan memilih konsentrasi yang sesuai dengan konsentrasi pilihannya dan portal tersebut mendeteksi konsentrasi yang berbeda dari konsentrasi yang diinginkan mahasiswa. Sedangkan sisanya merasa kesulitan memonitor informasi terbaru mengenai tugas akhir yang dimana kesulitan ini tidak dimasukkan dalam bagian identifikasi masalah.

Penelitian mengenai klasifikasi teks menggunakan web semantik telah dilakukan sebelumnya. Web semantik mengaktifkan komputer untuk memahami data. Alih-alih untuk membuat komputer untuk memahami manusia, web semantik membantu komputer untuk memecahkan masalah yang telah didefinisikan dengan data yang telah didefinisikan juga.

Dalam penggunaan web semantik tersebut, kita bisa mendapatkan salah satu manfaat penggunaannya, yaitu pengambilan informasi atau *information retrieval*. Web semantik dapat membantu kita untuk mendapatkan informasi atau jawaban yang kita butuhkan. Hal ini sesuai dengan konteks klasifikasi teks yang akan dikembangkan pada penelitian ini yang dimana kita dapat mendapatkan jawaban konsentrasi yang sesuai dengan masukan judul proposal tugas akhir. Selain itu, data yang telah didefinisikan sebelumnya dapat digunakan kembali untuk penelitian yang lain, dengan catatan untuk domain yang sama dan sangat memungkinkan untuk dilakukannya improvisasi menyesuaikan kebutuhan. Maka dari itu, penggunaan web semantik dirasa cocok untuk penelitian ini.

Berdasarkan uraian yang dijelaskan sebelumnya, penelitian ini bermaksud untuk mengembangkan program untuk mengklasifikasikan judul proposal tugas akhir mahasiswa menggunakan semantik web guna membantu dosen untuk memeriksa konsentrasi yang terkandung di dalamnya dengan judul penelitian ***“Klasifikasi Judul Tugas Akhir Menggunakan Teknologi Semantik Web”***.

1.2 Pertanyaan Penelitian

Pertanyaan penelitian yang menyebabkan adanya penelitian ini antara lain:

- a. Pendekatan teknologi semantik apakah yang dapat digunakan untuk mengklasifikasi teks judul proposal tugas akhir?
- b. Seberapa akuratkah hasil pendekatan tersebut?

1.3 Batasan Masalah

Untuk menjaga fokus penelitian dalam tugas akhir ini, perlu diperhatikan beberapa batasan masalah yang perlu diperhatikan:

- a. Studi kasus penelitian atau objek penelitian ini adalah proposal tugas akhir mahasiswa, khususnya di Jurusan Informatika FTI UIL.
- b. Domain yang diangkat adalah Informatika (*computer science*) dengan menyesuaikan Panduan Kurikulum 2010 Jurusan Informatika.

1.4 Tujuan Penelitian

Tujuan penelitian dari pertanyaan penelitian yang ada, yaitu:

- a. Pendekatan teknologi semantik yang digunakan.

- b. Mengetahui seberapa akurat penerapannya menggunakan pendekatan tersebut.

1.5 Manfaat Penelitian

Manfaat yang didapat dari penelitian ini, yaitu:

- a. Mempermudah dosen untuk memeriksa sebuah proposal tugas akhir dan menentukannya ke sebuah konsentrasi.
- b. Tidak terjadi kesalahan dalam mengunggah proposal tugas akhir yang tidak sesuai konsentrasinya oleh Mahasiswa.

1.6 Metode Penelitian

Langkah-langkah yang dilakukan diantaranya:

- a. Pengumpulan data. Tahap ini didetailkan menjadi dua bagian:
 - 1. Studi Literatur: Studi literatur dilakukan untuk mempelajari penelitian-penelitian sebelumnya yang memiliki latar belakang yang sama sehingga dapat menunjang penelitian yang dilakukan pada tugas akhir ini. Selain itu, informasi dari literatur-literatur pada penelitian sebelumnya dapat digali untuk ditelaah dan dapat dikembangkan atau diperbaiki di penelitian sekarang.
 - 2. Identifikasi Masalah: Tahap identifikasi masalah dilakukan dengan cara mewawancarai orang-orang yang pernah memakai Portal KPTA FTI UII, khususnya di Jurusan Informatika FTI UII. Diharapkan hasil dari tahap ini akan memberikan gambaran terkait pemakaian Portal KPTA FTI UII pada masa sekarang.
- b. Tahap Pengembangan. Tahap ini terdiri dari tiga bagian:
 - 1. Pra pemrosesan: Bagian ini berperan untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan selanjutnya (Februariyanti, 2012). Terdiri dari *case folding*, *stopword removal*, dan tokenisasi (Ayedh et al., 2016).
 - 2. Implementasi pendekatan semantik untuk klasifikasi teks: Tahap ini dimulai setelah teks judul telah melalui pra pemrosesan teks. Selanjutnya, setelah teks judul melalui proses tokenisasi (pada pra pemrosesan teks), akan dibuat kombinasi kata yang akan menghasilkan frasa. Frasa tersebut akan melalui proses kueri SPARQL pada ontologi yang telah dikembangkan sebelumnya. Hasil kueri frasa tersebut akan dihitung jumlah kelas yang keluar dan kelas yang paling banyak keluar akan menjadi kelas teks judul tersebut.

c. Pengujian dan Evaluasi

Tahap terakhir adalah pengujian pendekatan yang diajukan. Pendekatan tersebut akan diuji dengan beberapa teks dan menetapkan apakah pengujian ini berhasil dengan hasil keluaran yang sesuai atau tidak.

1.7 Sistematika Penulisan

Sistematika penulisan dibuat untuk memudahkan penulisan laporan akhir. Berikut ini sistematika pembahasan dalam penelitian ini:

BAB I PENDAHULUAN

Pada bab ini dijelaskan mengenai latar belakang, permasalahan, pertanyaan penelitian, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini berisi tentang semua hal yang berkaitan dengan klasifikasi teks, diantaranya *text preprocessing*, penjelasan teknologi semantik, dan penelitian yang pernah dilakukan sebelumnya atau tinjauan pustaka.

BAB III METODOLOGI PENELITIAN

Pada bab ini memuat rincian metode penelitian yang telah dibahas pada Bab I. Meliputi hasil dari identifikasi masalah, pengembangan ontologi, dan klasifikasi teks.

BAB IV HASIL DAN PEMBAHASAN

Berisi mengenai hasil dari implementasi pendekatan dan pengujian pendekatan.

BAB V KESIMPULAN DAN SARAN

Berisi mengenai kesimpulan akhir dari pendekatan yang telah dibuat dan saran untuk penelitian selanjutnya.

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Semakin berkembangnya internet, orang-orang semakin mudah untuk mengakses informasi. Berbagai bentuk informasi bisa didapatkan di internet, salah satunya adalah teks. Teks juga merupakan bentuk umum dari sebuah dokumen. Karena keberadaan dokumen-dokumen di internet semakin meningkat, maka penelitian mengenai klasifikasi dokumen maupun teks semakin meningkat juga dan menjadi sangat penting. Penelitian dengan topik ini cukup menantang karena semakin banyaknya dokumen yang tersedia, semakin banyak pula informasi relevan yang akan ditemukan (Qazi & Goudar, 2018).

Penelitian mengenai klasifikasi teks dengan menggunakan pendekatan web semantik telah dilakukan oleh peneliti beberapa tahun belakangan ini. Klasifikasi teks adalah mengelompokkan teks ke kategori yang telah ditentukan sebelumnya berdasarkan fitur yang menjadi patokan dari suatu kategori. Secara umum, klasifikasi teks termasuk peran penting dalam ekstraksi informasi dan peringkasan (*information extraction and summarization*), pengambilan teks (*text retrieval*), pendeteksi spam dalam surat elektronik, dan sebagainya (Jasser, 2015). Lain halnya, semantik artinya adalah mempelajari makna. Jika komputer dapat memahami makna dari sebuah teks, komputer tersebut tidak hanya menginterpretasi serangkaian karakter yang membentuk teks tersebut (kata maupun kalimat), tetapi juga membantu memisahkan makna teks dan data menggunakan teknologi tertentu (Krishnaraj et al., 2018).

Penelitian dalam bidang ini telah dilakukan dalam berbagai domain. Seperti penelitian pada domain pendakian gunung (Admojo & Winarko, 2016), yang dimana sistem pencarian informasi dikembangkan khusus untuk menyediakan informasi bagi para pendaki gunung di Indonesia dengan memanfaatkan dua ontologi sebagai teknologi semantiknya sebagai representasi data, yaitu ontologi bahasa dan ontologi *mountaineering* serta menggunakan fitur *spelling checker*. Pada penelitian yang serupa dalam domain sepak bola, ekstraksi informasi dan pengambilan informasi dengan representasi data yang sama, yaitu ontologi (Sabuncu et al., 2012). Penelitian tersebut menggunakan pendekatan *keyword-based semantic retrieval* dan *semantic indexing* serta menghasilkan peringkat berdasarkan tingkat kepentingan dengan memanfaatkan *library Apache Lucene*. Penelitian selanjutnya mengenai klasifikasi berita

digital dalam domain berita kriminal. Sebuah berita akan dikelompokkan ke dalam beberapa kelas kriminalitas berdasarkan hubungan di sekitar kelompok-kelompok itu sehingga proses klasifikasi dapat dilakukan secara sederhana dan spesifik (Rahma et al., 2019). Penelitian lainnya tentang klasifikasi dokumen dilakukan secara otomatis ke beberapa topik yang telah ditentukan secara dinamis (Allahyari et al., 2014). Dengan representasi data berupa ontologi, penelitian ini menggunakan ontologi domainnya menjadi *classifier* yang dimana klasifikasi topik diekspresikan menggunakan konteks ontologi yang telah dikembangkan.

Penelitian lain tentang klasifikasi dokumen dengan memanfaatkan kedekatan konsep. Pengklasifikasian dokumen dengan cara membandingkan dokumen ke masing-masing kategori dengan mengekstrak kata atau frasa dengan representasi data berupa *semantic hierarchy* dari WordNet (Peng & Choi, 2005). Sebaliknya, dengan representasi data yang berbeda, yaitu ontologi, kata-kata kunci dari dokumen akan dipetakan ke konsep yang ada di dalam ontologi domain dan memanfaatkan *machine learning* untuk melatih *classifier* (Kastrati, 2017).

Penelitian serupa pada domain *short message* atau *short text* (pesan pendek). Pesan pendek tersebut diperkaya melalui *multi-granularity topics* yang nantinya menghasilkan fitur-fitur (Chen et al., 2011). Pada penelitian lain, dengan memanfaatkan *semantic dictionary* sebagai representasi data sehingga tidak membutuhkan *training set* untuk proses klasifikasi (Hao-jin, 2013).

Klasifikasi dokumen dengan bahasa tertentu juga pernah dilakukan oleh para peneliti beberapa tahun belakangan ini. Seperti klasifikasi dokumen berbahasa Arab yang memanfaatkan ontologi (Jasser, 2015). Klasifikasi tersebut memanfaatkan anotasi pada dokumen untuk memetakan istilah ke kelas yang sesuai dan menempatkan dokumen ke kategori yang sesuai. Penelitian lain yaitu klasifikasi dokumen berbahasa Telugu memanfaatkan analisis istilah pada tingkat kalimat dan dokumen. Model ini berbasis konsep yang secara efektif dapat membedakan antara istilah yang tidak penting sehubungan dengan semantik kalimat dan istilah yang mengandung konsep yang mewakili makna kalimat (Durga & Govardhan, 2011).

Berdasarkan tinjauan pustaka yang telah dilakukan, mayoritas penelitian sebelumnya menggunakan ontologi sebagai teknologi semantik sekaligus representasi data. Ontologi terkenal dengan fleksibilitasnya dikarenakan dapat dikembangkan sendiri sesuai dengan kebutuhan. Oleh karena itu, pada penelitian ini sejalan dengan beberapa penelitian sebelumnya yang memanfaatkan ontologi. Selain itu, dengan memanfaatkan ontologi sebagai kunci klasifikasi teks, maka proses klasifikasi dapat dilakukan tanpa *classifier training* atau *training*

set seperti pada penelitian (Hao-jin, 2013) dan (Allahyari et al., 2014). Penelitian ini juga sejalan dengan penelitian (Peng & Choi, 2005) dengan cara mengekstrak kata-kata atau frasa-frasa penting di dalam teks.

Tabel 2. 1 merupakan daftar penelitian sebelumnya atau tinjauan pustaka yang telah dirangkum oleh penulis.

Tabel 2. 1 Tinjauan Pustaka

| No. | Peneliti | Fokus Peneliti | Pendekatan Teknologi Semantik | Hasil Penelitian |
|-----|---|---|---|--|
| 1. | Fadhila Tangguh Admojo dan Edi Winarko (2016) | Pengembangan sistem pencarian informasi untuk jalur pendakian gunung | Ontologi dan menggunakan <i>spelling checker</i> | <ul style="list-style-type: none"> • Ontologi bahasa dan ontologi mountaineering • Sistem pencarian informasi terkait pendakian gunung yang dengan keluaran teks dan peta |
| 2. | Soner Kara, Ozgur Alan, Orkunt Sabuncu, Samet Akpınar, Nihan K. Cicekli, dan Ferda N. Alpaslan (2011) | Ekstraksi informasi berbasis ontologi dan pengambilan informasi (<i>information retrieval</i>) di dalam domain sepak bola | Ontologi dan menggunakan <i>Keyword-based semantic retrieval</i> dan <i>semantic indexing</i> | <ul style="list-style-type: none"> • Ontologi sepak bola • <i>Semantic querying and ranking</i> |
| 3. | Xiaogang Peng dan Ben Choi (2005) | Klasifikasi dokumen berdasarkan arti kata dan hubungannya dengan konsep-konsep | <i>Semantic hierarchy</i> dari WordNet dan mengekstrak kata-kata di dalam dokumen dan digunakan untuk meningkatkan bobot <i>synset</i> di dalam WordNet serta mendefinisikan hubungan antar <i>synset</i> | Pengklasifikasian dokumen dengan cara membandingkan dokumen ke masing-masing kategori yang sudah ditentukan dengan mengekstrak fitur (kata atau frasa). Dokumen yang memiliki kemiripan dengan kategori tertentu akan dikategorikan ke kategori tersebut |

| | | | | |
|----|--|---|--|--|
| 4. | Zenun Kastrati dan Sule Yildirim Yayilgan (2017) | Klasifikasi dokumen berdasarkan konsep-konsep di ontologi | Ontologi dan <i>Mapping</i> kata-kata kunci dari dokumen ke konsep yang ada di ontologi domain dan memanfaatkan <i>machine learning</i> untuk melatih <i>classifier</i> dan untuk memprediksi model yang dapat mengklasifikasi dokumen yang belum terlabeli ke kategori tertentu | Model klasifikasi teks dokumen berdasarkan ontologi |
| 5. | Tang Hao-jin, Yan Dan-feng, dan Tian Yuan (2013) | Klasifikasi pada pesan pendek (<i>short message</i>) | <i>Semantic dictionary</i> dan membangun sebuah kamus domain dengan menganalisis karakteristik spesifik di bidang tertentu | Sebuah metode untuk mengklasifikasi teks pendek tanpa bantuan <i>training set</i> . Klasifikasi dengan metode ini sangat bergantung pada <i>semantic dictionary</i> yang dibuat dengan cara manual |
| 6. | Mehdi Allahyari, Krys J. Kochut, dan Maciej Janik (2014) | Klasifikasi dokumen teks secara otomatis ke beberapa topik yang ditentukan secara dinamis | Data dalam format RDF atau Ontologi dan mengukur kemiripan semantik dari <i>thematic graph</i> yang dibuat dari dokumen teks dan <i>ontology sub-graphs</i> yang dihasilkan dari proyeksi konteks yang ditentukan | Sebuah metode untuk mengklasifikasi dokumen dengan memanfaatkan ontologi pengetahuan dan tanpa <i>classifier training</i> atau <i>training set</i> |
| 7. | Mangen Chen, Xiaoming Jin, dan Dou Shen (2011) | Klasifikasi teks pendek (<i>short text</i>) | <i>Universal dataset</i> dan memperkaya teks pendek melalui <i>multi-granularity topics</i> yang nantinya akan menghasilkan fitur-fitur baru | Sebuah metode untuk mengklasifikasi teks pendek dengan menggunakan <i>multi-granularity topics</i> untuk menghasilkan fitur-fitur untuk teks pendek |

| | | | | |
|-----|--|---|--|--|
| 8. | Mohammed M. Abu Jasser (2015) | Klasifikasi dokumen bahasa Arab | Ontologi dan menggunakan anotasi, yaitu memetakan istilah ke kelas yang sesuai dan menempatkan dokumen ke kategori yang sesuai | Sebuah pendekatan baru untuk mengklasifikasi dokumen bahasa Arab dengan mengimplementasi ontologi untuk memberikan anotasi dokumen untuk diklasifikasi berdasarkan anotasinya dibandingkan dengan bergantung pada klasifikasi berdasarkan kata kunci |
| 9. | Rozilawati Binti Dollah dan Masaki Aono (2011) | Klasifikasi abstrak teks biomedis dengan pendekatan ontologi | Ontologi dengan mengeksplorasi penggunaan struktur 'konsep' hierarkis dengan bantuan algoritma penyesuaian ontologi untuk mencari dan mengidentifikasi kategori yang mungkin untuk mengklasifikasikan abstrak teks biomedis | Sebuah pendekatan klasifikasi hirarkis yang memanfaatkan algoritma penyesuaian ontologi untuk tujuan klasifikasi abstrak teks biomedis |
| 10. | A. Kanaka Durga dan A. Govardhan (2011) | Klasifikasi teks berbasis ontologi untuk dokumen berbahasa Telugu | Ontologi dan menganalisis istilah pada tingkat kalimat dan dokumen. Model berbasis konsep secara efektif membedakan antara istilah-istilah yang tidak penting sehubungan dengan semantik kalimat dan istilah-istilah yang mengandung konsep-konsep yang mewakili makna kalimat | Sebuah model klasifikasi dokumen berbahasa Telugu berbasis ontologi dengan model berbasis konsep |
| 11. | F. Rahma, D. D. Pangestu, A. Herdiani, dan N. | Klasifikasi berita digital kriminal | Ontologi dan menganalisis data untuk mendapatkan informasi tentang tren kriminalitas dengan | Ontologi dapat menjadi pengelompokan klasifikasi teks menjadi beberapa kelompok dengan memanfaatkan struktur |

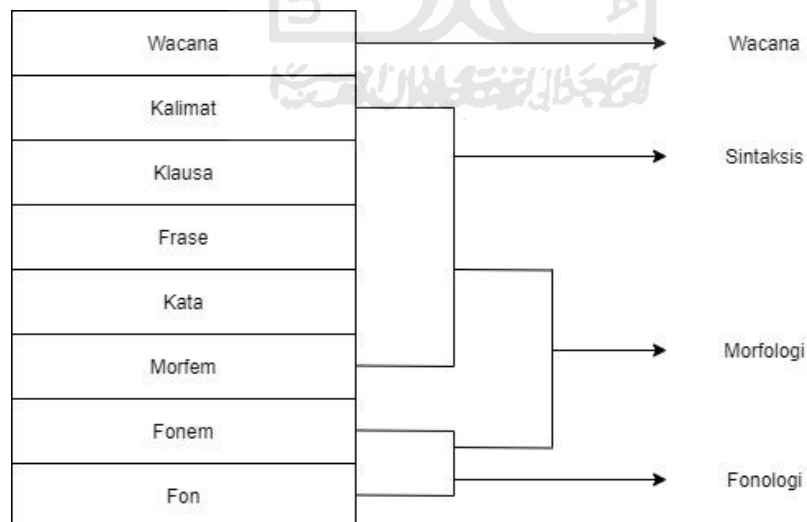
| | | | | |
|--|----------------------|----------------------|---|--|
| | Selviandro (2019) | berbasis ontologi | menggunakan metode klasifikasi teks berbasis ontologi | atau keterhubungan kelas dan contoh dalam desain ontologi |
|--|----------------------|----------------------|---|--|

2.2 Landasan Teori

2.2.1 Bahasa

Bahasa merupakan alat komunikasi antar makhluk hidup, tak terkecuali bagi manusia. Bahasa merupakan suatu sistem lambang bunyi yang bersifat arbitrer (tidak ada hubungan wajib antara kata atau leksem dengan benda atau konsep yang ditandai atau referen dari kata atau leksem tersebut) (Chaer, 1990). Seperti contoh kata [bunga] yang dilambangkan dengan benda atau konsep “bagian tumbuhan yang akan menjadi buah” menurut Kamus Besar Bahasa Indonesia (KBBI). Tidak dapat dijelaskan mengapa benda atau konsep tersebut dilambangkan dengan bunyi [bunga]. Jika ada hubungan wajib antara lambang dengan yang dilambangkannya, maka tidak akan ada bermacam-macam bahasa (P & Abdullah, 2013).

Bahasa juga merupakan sistem yang dimana terdiri atas unsur-unsur yang disusun. Secara hierarki, tataran bahasa dapat digambarkan seperti Gambar 2. 1 Hierarki Bahasa Gambar 2. 1.



Gambar 2. 1 Hierarki Bahasa

Sumber: P & Abdullah (2013)

Kata

Kata merupakan suatu unit paling kecil dalam bahasa yang mengandung konsep atau gagasan tertentu. Menurut Aristoteles (384-322 SM) adalah satuan terkecil yang mengandung makna (Chaer, 1990). Kata memiliki stabilitas intern dan mobilitas posisional, yang artinya ia memiliki komposisi tertentu (entah fonologis atau morfologis) dan secara relatif memiliki distribusi yang bebas (Keraf, 1984). Maksud distribusi yang bebas adalah seperti kalimat: *Saya makan nasi goreng itu; nasi goreng itu saya makan; kumakan nasi goreng itu.*

Menurut sumber yang sama, kata sebagai satuan dari perbendaharaan kata sebuah bahasa yang mengandung dua aspek, yaitu aspek bentuk atau ekspresi dan aspek isi atau makna. Aspek bentuk atau ekspresi adalah segi yang dapat diserap dengan pancaindra, yaitu dengan mendengar atau dengan melihat. Sebaliknya dengan aspek isi atau makna yang dimana menimbulkan reaksi dalam pikiran pendengar atau pembaca karena rangsangan aspek bentuk tadi. Reaksi yang timbul dapat berwujud sebagai “pengertian” atau “tindakan” atau bahkan keduanya.

Dalam berkomunikasi, tidak hanya berhadapan dengan “kata”, tetapi juga berhadapan dengan suatu “amanat” yang dijabarkan dengan beberapa unsur, yaitu: pengertian, perasaan, nada, dan tujuan. Pengertian merupakan landasan dasar untuk menyampaikan hal-hal tertentu kepada pendengar atau pembaca. Perasaan lebih mengarah kepada sikap pembicara terhadap apa yang dikatakannya. Nada mengarah ke sikap pembicara atau penulis kepada pendengar atau pembacanya. Sedangkan tujuan adalah efek yang ingin dicapai oleh pembicara atau penulis. Hal-hal tersebut adalah bagian dari memahami makna dalam komunikasi.

Frasa

Frasa adalah satuan gramatik atau gabungan terdiri dari dua kata atau lebih yang ukurannya lebih besar dari kata dan lebih kecil dari kalimat. Frasa bersifat nonpredikatif yang dimana bahwa salah satu kata dari gabungan tersebut bukan termasuk dalam golongan predikat. Sifat ini yang membedakan frasa dari klausa dan kalimat (Chaer, 2007).

Kalimat

Kalimat tersusun dari kata-kata. Kata-kata di dalam kalimat tersebut berkelompok dan membentuk satuan-satuan yang mempunyai fungsi tertentu yang disebut juga sebagai *gatra* (Hastuti, 1993). Kalimat juga merupakan satuan bahasa yang terdiri dari dua kata atau lebih

yang mengandung pikiran pokok dan dapat mengungkapkan suatu informasi secara lengkap (Sasangka, 2014).

Hal yang membedakan kalimat dengan klausa adalah jika terdapat sebuah tuturan yang menginformasikan sesuatu, akan tetapi belum utuh, tuturan tersebut belum bisa disebut dengan kalimat. Bisa jadi tuturan tersebut adalah frasa ataupun klausa (Sasangka, 2014). Seperti contoh di bawah ini:

- a. *Aku lapar* (klausa)
- b. *Maka aku makan* (klausa)
- c. *Aku lapar maka aku makan* (kalimat yang terdiri dari dua klausa)

Berbeda dengan jenis kalimat di bawah ini:

- a. *Buka pintunya!*
- b. *Jangan berdiri di sini!*
- c. *Waspadalah!*

Ketiga tuturan di atas merupakan kalimat karena sudah mengungkapkan pikiran pokok. Kelengkapan pikiran pokok ditunjukkan dengan adanya situasi pembicaraan serta alunan nada atau intonasi yang menyertainya.

Struktur inti kalimat dalam bahasa Indonesia umumnya berupa subjek dan predikat (S-P) (Sasangka, 2014). Struktur tersebut dapat diperluas menjadi beberapa tipe seperti di bawah ini:

- a. subjek-predikat (S-P)
- b. subjek-predikat-objek (S-P-O)
- c. subjek-predikat-keterangan (S-P-K)
- d. subjek-predikat-pelengkap (S-P-Pel)
- e. subjek-predikat-objek-pelengkap (S-P-O-Pel)
- f. subjek-predikat-keterangan (S-P-K)
- g. subjek-predikat-objek-keterangan (S-P-O-K)

2.2.2 NLP

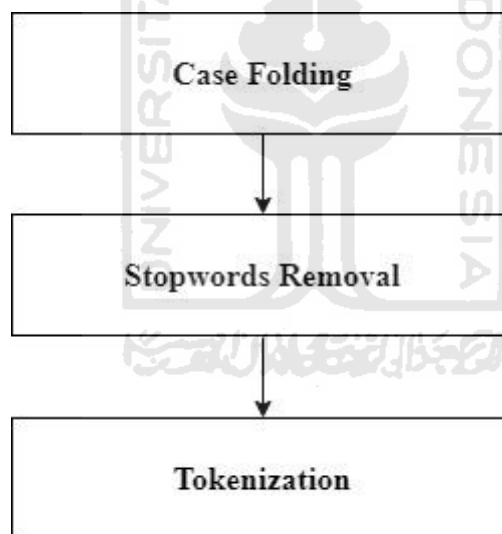
Menurut (Beysolow, 2018), *Natural Language Processing* (NLP) adalah cabang dari *computer science* atau informatika yang berfokus pada memungkinkan komputer atau mesin untuk memahami bahasa manusia dengan cara yang “alami” seperti yang dilakukan manusia. Kata “alami” bermaksud untuk membedakan ucapan dan tulisan manusia dari bahasa formal

seperti matematika atau notasi logika atau bahasa komputer (Jackson & Moulinier, 2002). Biasanya, NLP akan merujuk pada tugas-tugas seperti memahami sentimen teks (*sentiment analysis*), pengenalan ucapan (*speech recognition*), dan menghasilkan respons terhadap pertanyaan. Pra pemrosesan teks secara tradisional merupakan langkah penting bagi NLP.

Pra pemrosesan Teks (*Text Preprocessing*)

Pra pemrosesan teks atau *text preprocessing* merupakan langkah terpenting sebelum data diolah dan diperoleh informasi yang relevan serta memasuki tahap klasifikasi. Pra pemrosesan teks digunakan untuk mengekstraksi bagian yang menarik dan pengetahuan dari teks yang tidak terstruktur (Kannan & Gurusamy, 2015). Menurut sumber yang sama, ada dua tujuan dari pra pemrosesan teks, yaitu untuk mengurangi pengindeksan atau data dari ukuran teks dokumen dan untuk meningkatkan efisiensi dan efektivitas dari proses pengambilan informasi (*information retrieval*).

Proses pra pemrosesan teks dapat ditunjukkan dalam Gambar 2. 2.



Gambar 2. 2 Proses Pra Pemrosesan Teks

- a. *Case Folding* (Penyeragaman Huruf): Penyeragaman huruf berarti proses untuk membuat semua huruf di dalam teks menjadi seragam. Dalam penelitian ini, penyeragaman huruf diubah menjadi huruf kecil (*lowercase*).
- b. *Stopwords Removal* (Menghilangkan *Stopwords*): *Stopwords* merupakan kata-kata yang kurang penting karena tidak memiliki makna. Kata-kata kurang penting tersebut juga sering muncul di tiap kalimat. Contoh *stopwords* diantaranya “di”, “ke”, “dari”, “yang”, dan sebagainya. Selain itu, tanda baca juga dihilangkan.

- c. *Tokenization* (Tokenisasi): *Tokenization* adalah membagi urutan *string* atau teks menjadi beberapa bagian token atau kata.

2.2.3 Permutasi

Permutasi adalah penyusunan kembali objek-objek yang berbeda dari urutan sebenarnya atau penyusunan objek-objek yang diurutkan secara spesifik (Rosen, 2012). Permutasi juga memperhatikan urutan, yang dimana satu objek berbeda dengan objek lainnya walaupun unsur di dalamnya sama. Penerapan permutasi dalam proses klasifikasi bertujuan untuk menemukan frasa sebagai kunci yang ada di dalam ontologi. Selain itu, dengan memanfaatkan sifatnya yang memperhatikan urutan, probabilitas frasa-frasa yang muncul semakin banyak dan dapat dicari satu persatu.

Persamaan umum permutasi ditunjukkan pada persamaan (2.1).

$$P_k^n = \frac{n!}{(n-k)!}, n > k \quad (2.1)$$

Keterangan:

P = Permutasi

n = Jumlah elemen

k = Jumlah elemen yang akan disusun

2.2.4 Klasifikasi

Klasifikasi, pada dasarnya, menetapkan sebuah objek ke dalam suatu kelas yang telah didefinisikan sebelumnya (Kastrati, 2017). Penerapan klasifikasi sudah sangat luas, meliputi beberapa hal, seperti *sentiment classification*, klasifikasi email (*spam* dan *ham*), serta *topic labelling*. Klasifikasi juga merupakan salah satu tugas penting di *Natural Language Processing* (NLP) (Yang et al., 2016).

Dalam hal klasifikasi teks maupun dokumen, teks maupun dokumen tersebut diklasifikasikan dengan cara mengidentifikasi kata kunci-kata kunci yang terdapat dalam dokumen. Model ini biasa disebut dengan *Traditional Information Retrieval* (TIR) yang nantinya akan memberikan bobot pada kata kunci untuk setiap dokumen untuk mencerminkan kepentingan relatif dari kata kunci dalam dokumen (Fraihat, 2015). Proses klasifikasi secara terawasi biasanya memanfaatkan pembelajaran mesin atau pendekatan statistik. Sebagian besar dari mereka mempelajari definisi kategori dan membuat pengkategorisasi dari seperangkat

dokumen pelatihan yang sebelumnya diklasifikasikan ke dalam sejumlah kategori tetap. Metode tersebut diantaranya *Support Vector Machine*, *Naïve Bayes*, *Decision Tree*, dan sebagainya (Allahyari et al., 2014).

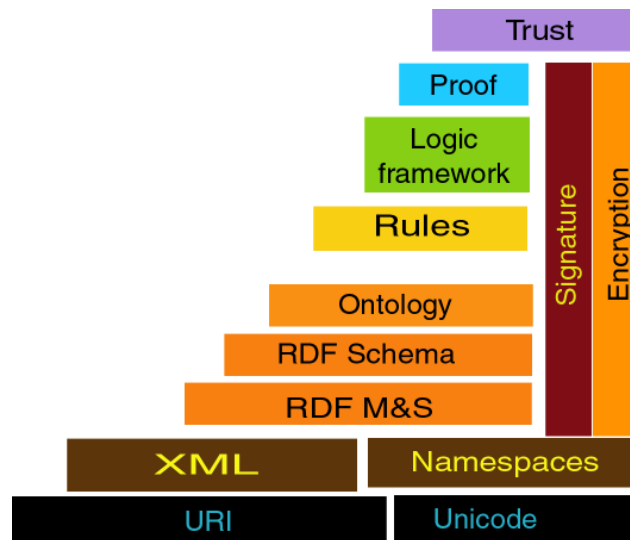
2.2.5 Web Semantik

Semantik berasal dari bahasa Yunani, *semantikos* yang artinya memberikan tanda. Semantik merupakan studi linguistik dan filosofis tentang makna dalam bahasa, bahasa pemrograman, logika formal, dan semiotik. Dengan kata lain, semantik berarti mempelajari tentang makna (Krishnaraj et al., 2018), baik itu yang melekat di tingkatan kata, frasa, kalimat, dan teks. Tidak hanya mempelajari makna, semantik juga mempelajari hubungan antara unit linguistik yang berbeda-beda, seperti polisemi, homonim, sinonim, antonim, dan sebagainya.

Pemanfaatan semantik dalam web dinamakan web semantik, yang mana pengembangan dari WWW atau biasa dikenal dengan *world wide web*. WWW yang ada saat ini tidak hanya menampilkan konten yang berbahasa format manusia, yang dapat dibaca oleh manusia, tetapi juga konten yang dapat dibaca oleh mesin. Web ini memiliki kecerdasan buatan yang membuatnya mampu untuk menemukan, menghubungkan, dan menggabungkan informasi di internet. Web semantik merevolusi web-web sebelumnya menjadi wadah untuk pertukaran data yang dapat menyesuaikan dengan keinginan pengguna sesuai kriteria yang ditentukan oleh pengguna itu sendiri. Selain dapat menghasilkan keluaran yang optimal, pengetahuan yang dimiliki oleh web akan diorganisir di ruang konseptual berdasarkan maknanya sehingga dapat mempercepat dan mengefisienkan klasifikasi, serta dapat beroperasi secara otomatis (Nandini, 2014).

Arsitektur Web Semantik

Web Semantik dibangun di atas Teknologi Semantik yang memiliki beberapa komponen pembangun, seperti Ontologi, RDF/OWL, SPARQL dan lainnya sehingga mesin dapat memahami data. Komponen-komponen tersebut ditunjukkan pada Gambar 2. 3.



Gambar 2. 3 Teknologi Web Semantik

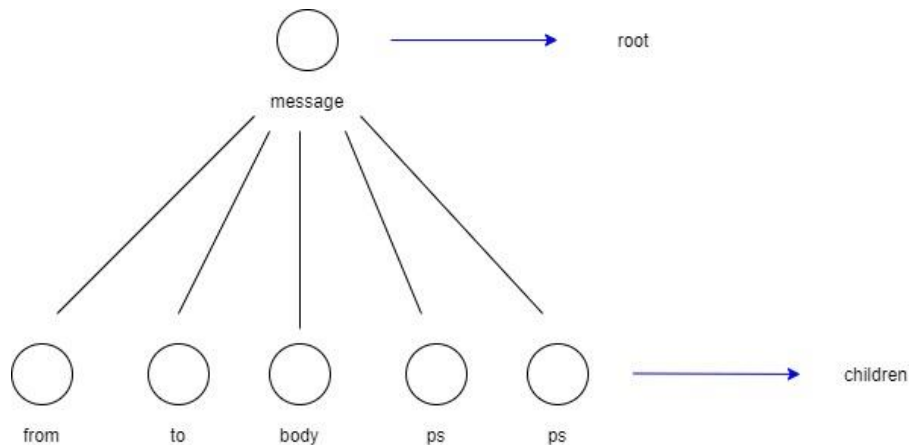
Sumber: w3c.org

XML

Extensible Markup Language atau XML berguna untuk mendeskripsikan data dan struktur-struktur mereka di dalam format yang dapat diakses oleh komputer dan untuk menyediakan arti standar untuk pertukaran data antar komputer. Tujuannya adalah untuk membuat internet cocok untuk komunikasi antar mesin (Szeredi, 2014).

Menurut sumber yang sama, dokumen XML adalah *file* teks yang dirancang untuk menyimpan data secara terstruktur. XML hanya sebagai alat penyimpanan yang ditujukan untuk menjadi format pertukaran data untuk sistem yang ingin “bekerja sama” sehingga terdapat komunikasi antar mesin.

Bagian terpenting dalam dokumen XML adalah elemen dan atribut (properti). Elemen XML terdiri dari tiga bagian: tag pembuka, data, dan tag penutup. Berdasarkan isinya, elemen XML bisa menjadi kompleks, bercampur, dan sederhana atau kosong. Elemen kompleks biasanya memiliki anak. Elemen sederhana hanya berisi data literal atau numerik. Elemen campuran mengandung elemen kompleks dan sederhana, sedangkan elemen kosong hanya terdiri dari tag pembuka dan tag penutup. Gambar 2. 4 merupakan contoh penggambaran elemen kompleks dengan satu akar dan lima anak.



Gambar 2. 4 Ilustrasi Elemen Kompleks XML

Sumber: Szeredi (2014)

Gambar 2. 5 adalah contoh penggambaran nyata XML dalam bentuk hierarki. Dalam contoh digambarkan terdapat seorang karyawan yang memiliki nama, gaji, nomor induk, dan nomor telepon. Format penulisan XML yang memiliki hubungan hierarki juga membuat XML cocok untuk mendeskripsikan struktur data.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<employees>
  <employee>
    <name>Jane Doe</name>
    <salary>4500</salary>
    <SSN>965342876</SSN>
  </employee>
  <employee>
    <name>John Stiles</name>
    <salary>6100</salary>
    <SSN>823457201</SSN>
    <phone>555-136-8531/2465</phone>
    <phone>917-555-7642</phone>
  </employee>
</employees>
  
```

Gambar 2. 5 Penulisan XML

Sumber: Szeredi (2014)

Selanjutnya adalah mengenai atribut (properti) XML. Atribut dalam XML dapat berubah-ubah, akan tetapi setiap atribut dapat muncul paling banyak sekali di setiap elemen. Atribut memiliki nama dan nilai yang dipisah dengan tanda sama dengan (=). Seperti pada Gambar 2. 6 yang dimana `from` dan `to` adalah bagian dari atribut.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<message from="Little Red Riding Hood" to="Granny">
  <body>I'll visit you this afternoon!</body>
  <ps>I'll bring some cookies</ps>
  <ps>they say there's a wolf in the forest</ps>
</message>

```

Gambar 2. 6 Penggunaan Atribut di dalam Elemen XML

Sumber: Szeredi (2014)

Elemen dan atribut termasuk pada golongan *local name*. *Local name* berfungsi sebagai nama. Namun, akan menjadi masalah saat menggabungkan beberapa dokumen XML, akan ada masalah nama yang saling bertentangan seperti pada Gambar 2.6. Maka dari itu, XML mendefinisikan *namespace* untuk menghindari konflik tersebut. Contohnya penggunaan *namespace* dalam dokumen XML tertanam pada halaman HTML dengan nama elemen `<body>` adalah seperti Gambar 2. 7.

```

<n:body> ... </n:body>

```

Gambar 2. 7 Contoh *Namespace* dalam XML

Sumber: Szeredi (2014)

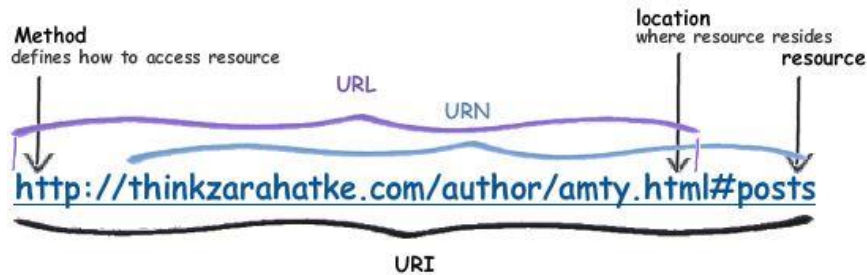
Sebuah *local name* diperpanjang dengan prefix yang dinamakan sebagai *qualified name* atau *universal name*. *Qualified name* merupakan nama yang unik sehingga bisa menghindari konflik kesamaan nama. Prefix secara tidak langsung memberi tahu mana *namespace* dari elemen yang diberikan.

URI

Unified Resource Identifier (URI) adalah sebuah harfiah atau arti sebenarnya dengan struktur yang terdefinisi dengan baik atau sebuah *resource*. URI berupa serangkaian karakter yang digunakan untuk mengidentifikasi objek-objek dan sumber atau nama-nama yang biasanya terdapat dalam web. URI juga mengidentifikasi sebuah sumber dari lokasi, nama, atau keduanya.

URI mempunyai peran penting di dunia web semantik. Properti yang terpenting dari URI adalah keunikannya, yang dimana memungkinkan adanya konstruksi pernyataan yang tidak ambigu. Perbedaannya dengan *Uniform Resource Locator* (URL) dan *Uniform Resource Name*

(URN) adalah URL merupakan bagian dari URI yang mengidentifikasi alamat, sedangkan URN mengidentifikasi nama. URL merupakan bagian dari URI, akan tetapi URI juga tidak selalu URL. Gambaran URI, URL, dan URN dapat dicontohkan oleh Gambar 2. 8.



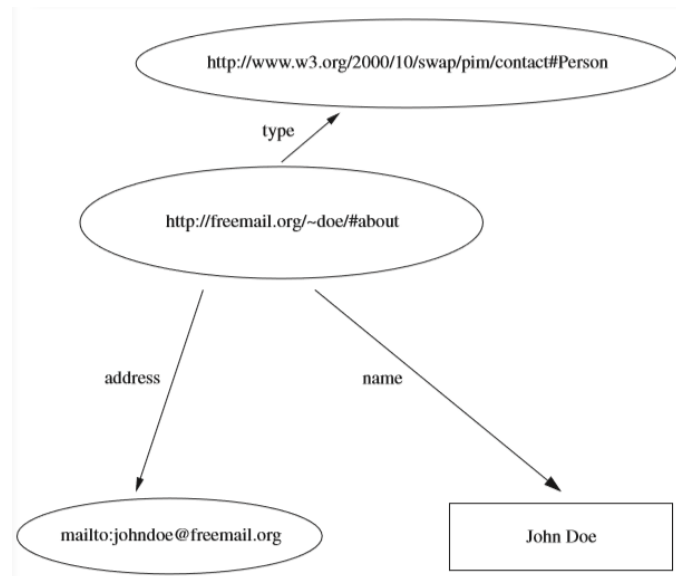
Gambar 2. 8 URI, URL, dan URN dalam Satu Kesatuan

Sumber: stackoverflow.com

RDF

RDF atau *Resource Development Framework* adalah format yang digunakan untuk menyimpan data di web semantik atau basis data *semantic graph* (Ontotext, n.d.) atau model data standar untuk pertukaran data di web (W3C, 2014). RDF memiliki fitur yang memfasilitasi penggabungan data dan memperluas struktur tautan dari web untuk menggunakan URI untuk memberi nama hubungan di antara satu objek dengan yang lain serta kedua ujung tautan, yaitu subjek dan objek. hal tersebut biasa disebut dengan *triple*. *Triple* juga merupakan data atomik dari RDF. Dengan menggunakan model sederhana ini, memungkinkan data terstruktur dan semi terstruktur untuk dicampur, diekspos, dan dibagikan di berbagai aplikasi.

Contoh *triple* dapat dituliskan pada kalimat berikut: “Alamat e-mail John Doe adalah johndoe@freemail.org”. Penggambaran *triple* menjadi RDF *graph* dapat diilustrasikan pada Gambar 2. 9.



Gambar 2. 9 Contoh RDF *Graph* yang Merepresentasikan *Triple*

Sumber: Szeredi (2014)

Ada sebuah *resource* yang teridentifikasi oleh URI `http://freemail.org/~doe/#about`. Di dalam RDF *graph* teridentifikasi sebagai *Person*, dijelaskan melalui URI `http://www.w3.org/2000/10/swap/pim/contact#Person` dengan *name* (nama) orang tersebut adalah John Doe dan memiliki *address* (alamat email) di dalam URI `mailto:johndoe@freemail.org`. Ketiga properti (*type*, *name*, *address*) sebenarnya juga diidentifikasi menggunakan URI.

Standar RDF didefinisikan dalam bentuk XML. Proses mengubah RDF *graph* menjadi deskripsi XML, atau rangkaian *triple*, disebut proses *linearisation* atau *serialisation* seperti contoh pada Gambar 2. 10.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://swexpld.org/utills#">

  <rdf:Description rdf:about=
    "http://freemail.org/~doe/#about">
    <s:name>John Doe</s:name>
    <s:address rdf:resource="mailto:johndoe@freemail.org"/>
    <rdf:type rdf:resource=
      "http://www.w3.org/2000/10/swap/pim/contact#Person"/>
  </rdf:Description>
</rdf:RDF>

```

Gambar 2. 10 RDF dalam format XML

Sumber: Szeredi (2014)

Penggunaan utama dalam RDF adalah untuk menyambungkan metadata dengan *resource*. Dengan kata lain, RDF adalah suatu kerangka kerja yang melampirkan semantik pada bagian-bagian informasi yang ditemukan di web. Spesifikasi RDF mendefinisikan model data RDF, yang dimana model data berdasarkan teori himpunan yang digunakan untuk menggambarkan metadata. Model tersebut diantaranya:

- a. *Resource*: diidentifikasi oleh URI. Contohnya seperti gambar yang ditemukan pada halaman web dan semua hal yang mempunyai URI.
- b. Properti: bagian dari *resource*. Fitur ini yang menempel pada *resource*. Dalam properti, dapat ditemukan *resource* apa yang dapat dilampirkan, nilai apa yang dapat diambil, dan hubungan apa terhadap properti lain.
- c. Literal: contohnya adalah urutan karakter.
- d. *Statement (triple)*: berisi subjek, predikat, dan objek. Subjek merupakan *resource* RDF yang berubah-ubah, predikat adalah properti RDF, sedangkan objek adalah nilai dari properti atau literal.

RDFS

Resource Description Framework Schema atau RDFS merupakan skema dari RDF atau merupakan perpanjangan semantik dari RDF yang menyediakan mekanisme untuk menggambarkan kelompok *resource* dan hubungan antar *resource* (W3C, n.d.). RDFS ditulis dalam sintaks RDF menggunakan *triple* dan dapat diakses menggunakan kueri SPARQL.

Menurut (W3C, n.d.), bagian-bagian dari RDFS adalah kelas, properti, dan utilitas. Penjabarannya adalah sebagai berikut:

- a. Kelas
1. *Resource* (*rdfs:Resource*): kelas dari segalanya. Semua hal dideskripsikan oleh RDF merupakan *resource*.
 2. Kelas (*rdfs:Class*): mendeklarasikan *resource* sebagai kelas untuk *resource* yang lain.
 3. Literal (*rdfs:Literal*): nilai literal seperti tipe data *string* dan *integer*.
 4. Tipe data (*rdfs:Datatype*): kelas dari tipe data dan merupakan subkelas dari *rdfs:Literal*.
 5. XMLLiteral (*rdfs:XMLLiteral*): kelas dari nilai literal XML yang juga merupakan *instance* dari *rdfs:Datatype* dan *rdfs:Literal*.
 6. Properti (*rdfs:Property*): kelas dari properti.
- b. Properti. Bagian ini mendeskripsikan hubungan dari *resource* subjek dan *resource* objek atau biasa disebut sebagai predikat.
1. *Domain* (*rdfs:domain*): berfungsi sebagai subjek di dalam sebuah *triple*.
 2. *Range* (*rdfs:range*): berfungsi sebagai objek di dalam sebuah *triple*.
 3. Tipe (*rdfs:type*): properti yang digunakan untuk menyatakan sebuah *resource* adalah sebuah *instance* dari kelas.
 4. *Subclass of* (*rdfs:subClassOf*): mendeklarasikan sub kelas dari kelas.
 5. *Subproperty of* (*rdfs:subPropertyOf*): sebuah *instance* dari *rdf:Property* yang digunakan untuk menyatakan bahwa semua *resource* yang terkait dengan satu properti juga terkait dengan yang lain.
 6. Label (*rdfs:label*): merupakan *instance* dari *rdf:Property* yang bisa dimanfaatkan untuk membuat *resource* yang dapat dibaca oleh pengguna.
 7. *Comment* (*rdfs:comment*): merupakan *instance* dari *rdf:Property* yang dapat digunakan untuk membuat deskripsi yang dapat dibaca oleh manusia.
- c. Utilitas. Bagian ini dibangun berdasarkan kosa kata RDF yang terbatas.
1. *See also* (*rdfs:seeAlso*): sebuah *instance* dari *rdf:Property* yang dapat digunakan untuk menunjukkan *resource* yang mungkin dapat menunjukkan informasi tambahan tentang *resource* subjek.
 2. *Is defined by* (*rdfs:isDefinedBy*): sebuah *instance* dari *rdf:Property* yang dapat digunakan untuk menunjukkan *resource* yang mendefinisikan *resource* subjek.

OWL

Web Ontology Language atau OWL merupakan bahasa web semantik yang dirancang untuk mewakili pengetahuan yang kaya dan kompleks tentang hal-hal, kelompok sesuatu, dan hubungan antara hal-hal tersebut. OWL menggunakan bahasa berbasis logika komputasi sehingga pengetahuan yang diekspresikan OWL dapat dieksploitasi oleh program komputer. Dokumen OWL, yang biasa disebut ontologi, dapat dipublikasikan di *World Wide Web* (WWW) dan dapat dirujuk serta merujuk dari ontologi lainnya (W3C, 2013). W3C, komunitas internasional yang merilis OWL, merekomendasikan OWL untuk kasus-kasus dimana kemampuan RDF tidak memadai (Szeredi, 2014).

OWL mirip dengan RDFS yang dimana dapat dianggap sebagai satu set URI. URI dalam set ini mengidentifikasi *resource* yang membantu membuat sistem terminologis yang menggunakan deskripsi RDF biasa dan dapat digunakan untuk menggambarkan *equivalence* atau *disjointness* dalam kelas-kelas, kardinalitas properti, dan sebagainya (Szeredi, 2014). Contoh penggunaan OWL dalam dokumen RDF terdapat pada Gambar 2. 11.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF
  xml:base = "http://swexpld.org#"
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl = "http://www.w3.org/2002/07/owl#">
  <owl:Class rdf:ID="Human" />
  <owl:Class rdf:ID="Female">
    <rdfs:subClassOf rdf:resource="#Human" />
  </owl:Class>
  <owl:ObjectProperty rdf:ID="hasChild">
    <rdfs:domain rdf:resource="#Human" />
    <rdfs:range rdf:resource="#Human" />
  </owl:ObjectProperty>
  <owl:Class rdf:ID="FatherOfGirls">
    <owl:intersectionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Human" />
      <owl:Class>
        <owl:complementOf rdf:resource="#Female" />
      </owl:Class>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#hasChild" />
        <owl:allValuesFrom rdf:resource="#Female" />
      </owl:Restriction>
    </owl:intersectionOf>
  </owl:Class>
</rdf:RDF>

```

Gambar 2. 11 Contoh OWL pada Dokumen RDF/XML

Sumber: Szeredi (2014)

Pada contoh di atas terdapat properti `owl:complementOf` yang dimana dapat dideskripsikan dalam skema RDF (RDFS) seperti pada Gambar 2. 12.

```

<rdf:Property rdf:ID="complementOf">
  <rdfs:label>complementOf</rdfs:label>
  <rdfs:domain rdf:resource="#Class"/>
  <rdfs:range rdf:resource="#Class"/>
</rdf:Property>

```

Gambar 2. 12 Deskripsi RDFS pada properti owl:complementOf

Sumber: Szeredi (2014)

Ini menunjukkan bahwa properti owl:complementOf menyambungkan kelas-kelas OWL dengan kelas-kelas OWL lainnya. Walaupun pada OWL memiliki versi RDFS-nya sendiri, ekspresivitas terbatas dari bahasa RDFS tidak memungkinkan seseorang untuk menangkap arti dari konstruksi OWL yang berbeda. Seperti contoh di atas yang tidak mengungkapkan bahwa complementOf menggambarkan komplemen kelas.

Bahasa OWL juga memungkinkan kita untuk mendeskripsikan *instance*. Mengikuti pada contoh sebelumnya, Gambar 2. 13 menunjukkan *instance* dari rdf:ID="Human" dan rdf:ID="FatherOfGirls" dalam bentuk RDFS. *Instance* dari Human dan FatherOfGirls adalah "Bob" dan "Ryan", sedangkan FatherOfGirls mempunyai properti hasChild yang menghubungkan dengan *instance* "Annie" yang jika dirangkai menjadi kalimat berarti *Ryan adalah ayah dari anak perempuan (father of girls) dan Annie adalah anak dari Ryan (Ryan has child Annie)*.

```

<Human rdf:ID="Bob" />

<FatherOfGirls rdf:ID="Ryan">
  <hasChild rdf:resource="#Annie" />
</FatherOfGirls>

```

Gambar 2. 13 Deskripsi RDFS pada *instance* Human dan FatherOfGirls

Sumber: Szeredi (2014)

SPARQL

SPARQL Protocol and the RDF Query Language atau disingkat SPARQL direkomendasikan oleh W3C pada tahun 2008. Konsep SPARQL mirip seperti *Structured Query Language* atau SQL pada basis data relasional yang mampu mengambil dan

memanipulasi data yang tersimpan pada format RDF. SPARQL memungkinkan untuk kueri yang berisi pola *triple*, konjungsi, disjungsi, dan pola-pola opsional (Cover, 2006).

Bentuk-bentuk kueri dalam SPARQL dalam membaca data pada format RDF dibedakan menjadi empat variasi yang berbeda untuk tujuan yang berbeda.

- a. SELECT, digunakan untuk mengekstrak nilai dari SPARQL yang nantinya menjadi hasil yang akan dimunculkan pada format tabel. Bentuk ini paling umum digunakan.
- b. CONSTRUCT, digunakan untuk mengekstrak informasi dari *endpoint* SPARQL dan mengubahnya menjadi RDF yang valid.
- c. ASK, digunakan untuk menyediakan hasil *boolean* (benar atau salah) untuk kueri pada *endpoint* SPARQL.
- d. DESCRIBE, digunakan untuk mengekstrak RDF *graph* dari *endpoint* SPARQL, tergantung informasi mana yang berguna yang dipilih oleh pengguna.

Masing-masing menggunakan kueri WHERE untuk membatasi kueri, kecuali pada DESCRIBE yang tidak wajib menggunakan WHERE. Gambar 2. 14 adalah contoh pencarian data dari RDF menggunakan SPARQL.

```

PREFIX s: <http://swexpld.org/util#>
SELECT ?x ?title
WHERE
{
  ?x s:publication ?pub .
  ?pub s:title ?title .
  FILTER regex(?title, "logic", "i")
}

```

Gambar 2. 14 Implementasi SPARQL

Sumber: Szeredi (2014)

Secara kalimat, kueri di atas mencari informasi “*siapa yang mempublikasi buku dengan judul dengan kata ‘logic’ di dalamnya?*”. Penjelasan kueri adalah sebagai berikut.

Namespace prefix adalah *s* dan *namespace URI* adalah *<http://swexpld.org/util#>*. Fungsi SELECT *?x ?title* mendeskripsikan bahwa hasil yang ingin ditampilkan adalah *x* dan *title* dalam format tabel. Fungsi WHERE menjelaskan tiga deskripsi:

- a. *?x s:publication ?pub* yang menjelaskan *triple* “*x melakukan publikasi*”.
- b. *?pub s:title ?title* yang menjelaskan bahwa “*publikasi memiliki judul*”.

- c. `FILTER regex (?title, "logic", "i")` yang menggunakan fungsi `FILTER` untuk menyaring judul yang mengandung kata *“logic”*. Argumen `i` digunakan untuk mendeklarasikan bahwa kata yang dikandung adalah *case-insensitive*. Fungsi `REGEX` adalah untuk mengembalikan `TRUE` saat variabel `?title` mengandung kata *“logic”*. Hasil yang ditampilkan adalah seperti pada Gambar 2. 15.

| x | title |
|---------------------------|---------------------------|
| <http://.../person#Peter> | "Logic-based ontology..." |

Gambar 2. 15 Hasil kueri SPARQL

Sumber: Szeredi (2014)

Dalam tabel hasil di atas mengungkapkan bahwa `x` yang merupakan orang bernama Peter (terlihat dalam URI yang ditampilkan) mempublikasi buku yang berjudul *Logic-based ontology*.

2.2.6 Ontologi

Ontologi merupakan model dimana hubungan yang secara eksplisit diberi nama dan dapat dibedakan (TopQuadrant Technology Briefing, 2003). Ontologi juga menjelaskan konsep dari sebuah domain tertentu yang mendefinisikan konsep domain tersebut (Nandini, 2014). Model tersebut menjadi jaringan hubungan antara satu kata dengan yang lain yang mengarah ke konsep tersebut (TopQuadrant Technology Briefing, 2003). Ontologi mendefinisikan kosakata umum untuk peneliti yang perlu berbagi informasi dalam domain. Ini mencakup definisi konsep dasar yang dapat ditafsirkan mesin dalam domain dan hubungan di antara mereka (Noy & McGuinness, 2000).

Ada beberapa alasan mengapa ontologi perlu dikembangkan menurut Noy dan McGuinness dalam makalahnya (Noy & McGuinness, 2000):

- Agar dapat berbagi pemahaman umum tentang struktur informasi: Hal ini merupakan salah satu tujuan yang lebih umum dalam mengembangkan ontologi. Contohnya, beberapa situs web yang berbeda berisi informasi medis atau menyediakan layanan *e-commerce* medis. Jika situs web ini berbagi dan mempublikasikan ontologi yang mendasari yang sama dari istilah yang mereka gunakan, maka agen komputer dapat mengekstraksi dan

mengumpulkan informasi dari situs yang berbeda ini. Agen dapat menggunakan informasi agregat ini untuk menjawab pertanyaan pengguna atau sebagai input data ke aplikasi lain.

- b. Memungkinkan penggunaan kembali pengetahuan domain: Sebuah ontologi dapat digunakan kembali, dengan catatan domain yang digunakan adalah sama. Selain itu, jika perlu membangun ontologi yang besar atau luas cakupan domainnya, dapat juga mengintegrasikan beberapa ontologi yang ada yang menggambarkan bagian-bagian dari domain besar.
- c. Membuat asumsi domain secara eksplisit: Ontologi dapat memberikan struktur hierarki dari istilah-istilah guna menjelaskan sebuah domain dan bagaimana relasi diantaranya.
- d. Menganalisis pengetahuan domain: Analisis formal terhadap istilah sangat penting ketika mencoba untuk menggunakan kembali ontologi yang ada dan memperluasnya.

Komponen Ontologi

Ontologi memiliki beberapa komponen yang dapat menjelaskan domain (Horridge, 2009), diantaranya:

- a. *Classes*, menjelaskan sebuah konsep dari domain yang terdiri dari beberapa *instance* atau individual. *Class* bisa disebut juga dengan *concepts* dan *categories*. *Class* terdiri dari *superclass* atau kelas utama dan *subclass* atau subkelas yang mana menjelaskan lebih spesifik dari *superclass*.
- b. *Instances*, atau disebut juga individual, yaitu anggota dari *class*. *Instances* juga dapat dianggap sebagai objek dari domain yang dibahas. Contoh *instances* terdapat pada Gambar 2. 16.

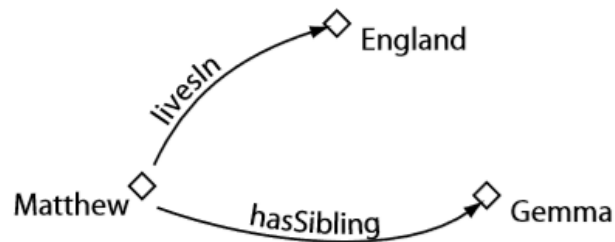


Gambar 2. 16 *Instances*

Sumber: Horridge (2009)

- c. *Properties*, atau disebut juga *Slots*, yang menjelaskan properti. Terdiri dari dua jenis, yaitu *object properties* dan *datatype properties*. *Object properties* akan menghubungkan *instance* dengan *instance* atau *subclass* dengan *subclass*, sedangkan *datatype properties*

menghubungkan *instance* dengan *datatype value*. Contoh properti terdapat pada Gambar 2. 17.

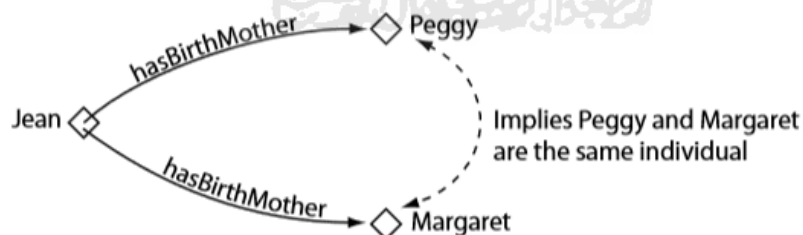


Gambar 2. 17 Properti

Sumber: Horridge (2009)

Ada beberapa jenis *properties*, diantaranya:

- a. *Functional Properties*: Menyatakan bahwa untuk setiap individu tertentu, properti dapat memiliki paling banyak satu nilai. Dengan kata lain, satu individu hanya bisa memiliki satu hubungan dengan individu yang lain. Jika beberapa individu ditentukan sebagai nilai untuk properti maka nilai-nilai ini akan disimpulkan untuk menunjukkan objek yang sama. Nama lain dari properti ini adalah *single valued properties* dan *features*. Seperti yang ditunjukkan pada Gambar 2. 18 menunjukkan individu **Jean hasBirthMother Peggy** dan **Jean hasBirthMother Margaret**. Properti **hasBirthMother** merupakan *functional property* yang mana kesimpulannya Peggy dan Margaret adalah individu yang sama karena Jean hanya memiliki satu *BirthMother*.

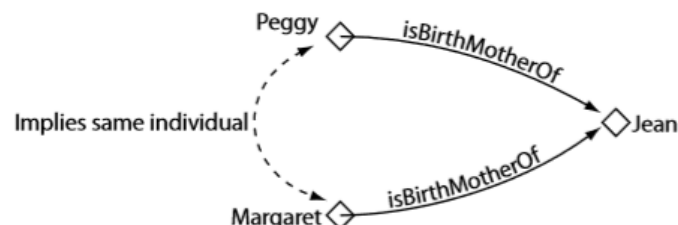


Gambar 2. 18 *Functional Properties*

Sumber: Horridge (2009)

- b. *Inverse Functional Properties*: Properti *inverse functional* berarti properti ini juga termasuk dalam *functional property*, yaitu satu individu hanya bisa memiliki satu hubungan dengan individu yang lain. Jika beberapa individu ditentukan sebagai nilai masuk untuk properti maka nilai-nilai ini akan disimpulkan untuk menunjukkan objek yang sama. Seperti yang ditunjukkan pada Gambar 2. 19 terdapat **Peggy isBirthMotherOf**

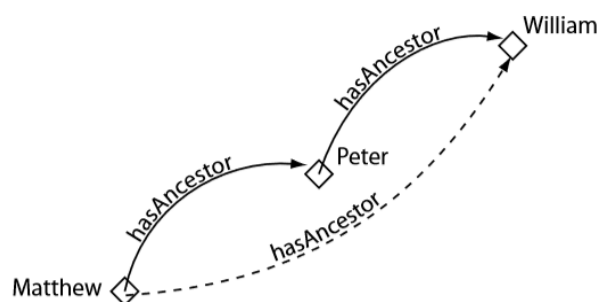
Jean dan **Margaret isBirthMotherOf Jean**. Properti yang disebutkan adalah **isBirthMotherOf** yang mana properti tersebut *inverse property* dari **hasBirthMother**. Kesimpulan dari contoh ini adalah Peggy dan Margaret, merupakan satu individu yang sama, sama-sama ibu dari Jean (**isBirthMotherOf**).



Gambar 2. 19 *Inverse Functional Properties*

Sumber: Horridge (2009)

- c. *Transitive Properties*: Properti ini mengartikan bahwa jika individu x terkait dengan individu y , dan individu y terkait dengan individu z , maka individu x akan terkait dengan individu z . Dengan kata lain, "lompatan" tunggal tersirat dalam rantai dua sepanjang properti yang diberikan jika properti itu transitif. Seperti pada Gambar 2. 20 yang menunjukkan **Matthew hasAncestor Peter** dan **Peter hasAncestor William**. Properti **hasAncestor** merupakan *transitive properties* yang artinya Matthew juga memiliki nenek moyang William menggunakan properti **hasAncestor**.

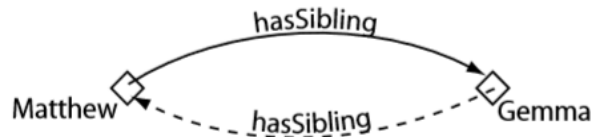


Gambar 2. 20 *Transitive Properties*

Sumber: Horridge (2009)

- d. *Symmetric Properties*: Properti ini mengkondisikan bahwa properti tersebut membuat kedua individunya saling terkait. Jadi, jika individu x terkait dengan individu y maka individu y juga harus terkait dengan individu x di sepanjang properti yang sama. Seperti

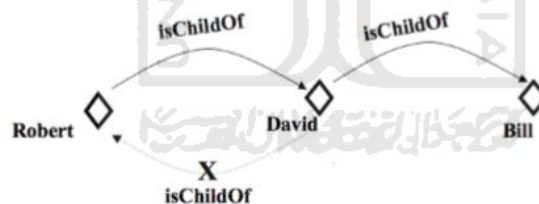
yang ditunjukkan pada Gambar 2. 21 yang mana **Matthew hasSibling Gemma**. Properti **hasSibling** merupakan *symmetric property* yang mana individu Gemma juga harus terkait dengan individu Matthew. Kesimpulannya, Matthew memiliki saudara bernama Gemma dan sebaliknya, Gemma memiliki saudara bernama Matthew, yang mana keduanya adalah bersaudara (sama-sama **hasSibling**). Artinya, properti ini bekerja untuk dua arah.



Gambar 2. 21 *Symmetric Properties*

Sumber: Horridge (2009)

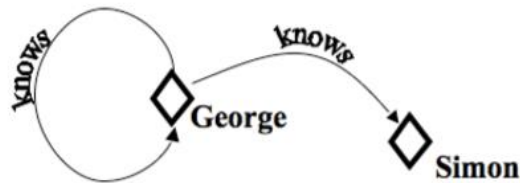
- e. *Asymmetric Properties*: Properti ini menunjukkan bahwa jika individu x terkait dengan individu y maka individu y tidak terkait dengan individu x di sepanjang properti yang sama. Seperti contoh yang ditunjukkan pada Gambar 2. 22, **Robert isChildOf David** yang berarti Robert adalah anak dari David. Tapi, properti **isChildOf** tidak dapat digunakan sebaliknya, yaitu David kepada Robert. Artinya, properti ini hanya bekerja untuk satu arah.



Gambar 2. 22 *Asymmetric Properties*

Sumber: Horridge (2009)

- f. *Reflexive Properties*: Properti ini menyatakan bahwa suatu properti bersifat refleksif yang menyebabkan setiap individu terkait dengan dirinya sendiri melalui properti itu. Seperti pada contoh yang terdapat di Gambar 2. 23, George memiliki properti **knows** yang dimaksudkan bahwa George dapat mengetahui dirinya sendiri. Selain itu, George juga dapat mengetahui orang lain, yaitu Simon, menggunakan properti yang sama.



Gambar 2. 23 *Reflexive Properties*

Sumber: Horridge (2009)

- g. *Irreflexive Properties*: Properti ini menyatakan bahwa suatu properti tidak bersifat refleksif, yang artinya seorang individu tidak dapat dikaitkan dengan dirinya sendiri melalui properti tersebut. Seperti yang ditunjukkan pada Gambar 2. 24, Alice memiliki properti **isMotherOf** yang menghubungkannya dengan Bob. Namun, properti tersebut tidak bisa dihubungkan dengan Alice sendiri.



Gambar 2. 24 *Irreflexive Properties*

Sumber: Horridge (2009)

Metode Pengembangan Ontologi

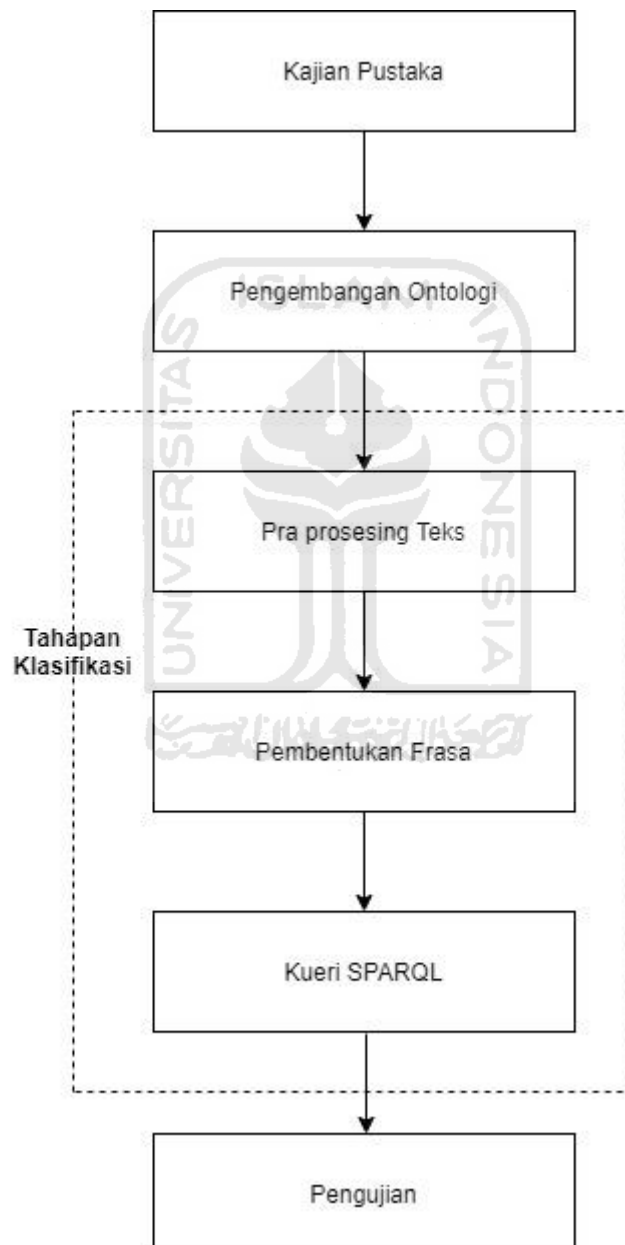
Pengembangan ontologi terdiri dari beberapa tahap menurut (Noy & McGuinness, 2000) dan (Jasser, 2015), yaitu:

- a. Mendefinisikan domain
- b. Mempertimbangkan untuk menggunakan ontologi yang telah ada
- c. Mendefinisikan kelas-kelas dalam hierarki taksonomi
- d. Mendefinisikan *slot* (properti)
- e. Mendefinisikan *facet* (batasan) pada *slot*
- f. Mendefinisikan *instance* (individu)

BAB III METODOLOGI PENELITIAN

3.1 Langkah-langkah Penelitian

Secara keseluruhan, langkah-langkah pada penelitian ini dapat digambarkan melalui bagan pada Gambar 3. 1.



Gambar 3. 1 Bagan Langkah-langkah Penelitian

3.2 Deskripsi Langkah-langkah Penelitian

Berdasarkan bagan pada Gambar 3. 1 akan dijabarkan langkah-langkah penelitian di bawah ini.

3.2.1 Kajian Pustaka

Langkah ini bertujuan untuk mengetahui penelitian sebelumnya yang serupa, baik itu secara pendekatan yang digunakan atau domain yang dikerjakan, sebagai panduan dalam melakukan penelitian ini. Literatur penelitian untuk kajian pustaka didapatkan dari jurnal penelitian dan situs-situs pencarian literatur di internet. Dalam proses kajian pustaka, penulis menelaah fokus peneliti, representasi data yang digunakan, pendekatan yang dilakukan, serta hasil penelitian. Selain itu, penulis juga mencoba memahami konsep klasifikasi teks maupun dokumen menggunakan teknologi web semantik. Representasi data dalam penelitian ini adalah menggunakan Ontologi dengan pendekatan mengekstraksi kata kunci yang terdapat teks judul tersebut dan dilakukan pencarian menggunakan kueri SPARQL.

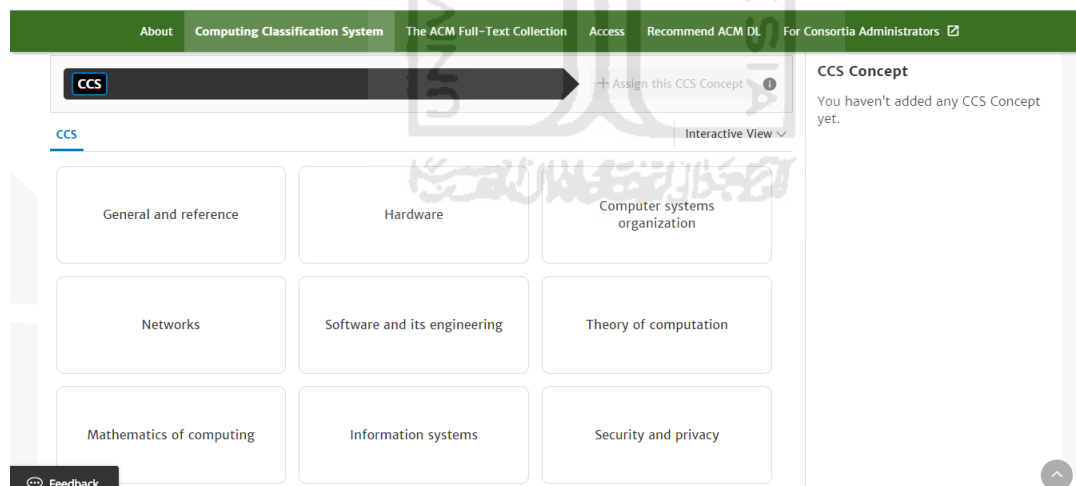
Penelitian ini sejalan dengan penelitian pada klasifikasi tugas akhir pada Informatika UII yang dimana melakukan klasifikasi judul tugas akhir dengan menggunakan algoritma *Naive Bayes* (Sembada, 2018). Klasifikasi tersebut membutuhkan bantuan *data training* agar mesin dapat mempelajari dan dapat memberi tahu konsentrasi tugas akhir tersebut. Namun, hal ini akan menjadi sulit ketika jumlah *data training* sedikit sehingga mempersulit proses klasifikasi. Dengan menggunakan pendekatan teknologi web semantik, proses klasifikasi tidak membutuhkan *data training* dan diharapkan klasifikasi teks judul tugas akhir dapat berjalan optimal. Maka dari itu, penggunaan web semantik dinilai cocok digunakan sebagai pendekatan pada penelitian ini.

3.2.2 Pengembangan Ontologi

Berdasarkan metode pengembangan ontologi yang tercantum pada (Noy & McGuinness, 2000), ontologi pada penelitian ini dikembangkan secara bertahap dengan deskripsi di bawah ini:

- a. Mendefinisikan domain: Dalam penelitian ini, ontologi yang dikembangkan berfokus pada domain Informatika (*computer science*). Di bawah domain Informatika terdapat beberapa *subclass* yang berkaitan dengan konsentrasi-konsentrasi yang ada di Informatika UII, yaitu Komputasi dan Sistem Cerdas, Multimedia Visi Komputer, Informatika Medis, Sistem Informasi, Jaringan dan Keamanan Komputer, dan Rekayasa Perangkat Lunak.

- b. Mempertimbangkan untuk menggunakan ontologi yang telah ada: Langkah ini untuk memastikan apakah ada ontologi yang dikembangkan sebelumnya di bidang studi yang sama (Jasser, 2015). Jika terdapat ontologi yang pernah dibuat untuk penelitian yang serupa, maka akan lebih mudah untuk memodifikasinya sesuai dengan kebutuhan. Pada penelitian ini, acuan pengelompokkan kelas yang digunakan pada ontologi ini berasal dari ontologi *Association for Computing Machinery* (ACM). Ontologi ACM merupakan ontologi utama untuk domain Informatika ini. Namun, perlu adanya modifikasi (menyesuaikan dengan panduan kurikulum) sehingga dibutuhkan ontologi lain untuk saling melengkapi. Ontologi lain yang dimaksud adalah diantaranya *Computer Science Ontology* (CSO), *Medical Subject Headings* (MeSH), *National Cancer Institute*, dan *International Classification of Disease Version 10* (ICD10). Tiga ontologi terakhir (MeSH, NCI, dan ICD10) difokuskan untuk membangun pengetahuan konsentrasi Informatika Medis. Ontologi dikembangkan berasal dari keempat sumber ontologi tersebut yang dimodifikasi dengan menyesuaikan konsentrasi-konsentrasi yang ada di Jurusan Informatika UII, khususnya pada Panduan Kurikulum 2010. Salah satu sumber ontologi yaitu ACM dapat ditunjukkan pada Gambar 3. 2.



Gambar 3. 2 Gambar Tampilan Website *Association for Computing Machinery* (ACM)

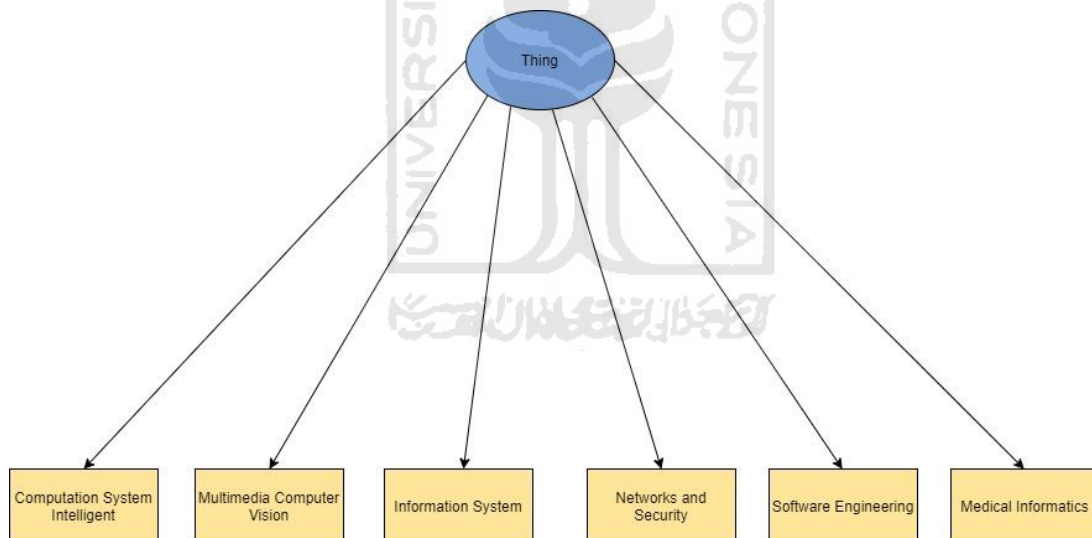
Sumber: dl.acm.org

Tabel 3. 1 merupakan konsentrasi yang terdapat pada Jurusan Informatika UII dengan dua bahasa yang berbeda, yaitu bahasa Indonesia dan bahasa Inggris.

Tabel 3. 1 Daftar Konsentrasi pada Informatika UII

| |
|---|
| Komputasi dan Sistem Cerdas (<i>Computational and Intelligent System</i>) |
| Multimedia Visi Komputer (<i>Multimedia Computer Vision</i>) |
| Informatika Medis (<i>Medical Informatics</i>) |
| Sistem Informasi (<i>Information Systems</i>) |
| Jaringan dan Keamanan Komputer (<i>Networks and Security</i>) |
| Rekayasa Perangkat Lunak (<i>Software Engineering</i>) |

- c. Mendefinisikan kelas-kelas dalam hierarki taksonomi: *Superclass* dari ontologi ini adalah *Computer Science* atau Informatika. Dari Informatika memiliki enam *subclass* dengan hierarki yang ditunjukkan pada Gambar 3. 3.

Gambar 3. 3 Taksonomi dari Domain *Computer Science*

Pembentukan taksonomi di atas menggunakan pendekatan *Top-Down*, yang mana proses pengembangan dimulai dari definisi yang paling umum di sebuah domain.

- d. Mendefinisikan *slot* (properti): Pada ontologi yang dikembangkan, properti yang digunakan adalah *subClassOf*. Seperti pada Gambar 3. 4 yang dimana kelas **Layering** merupakan subkelas dari kelas **Network Design Principles** yang ditunjukkan dengan *namespace* `rdfs:subClassOf`.

```

<owl:Class rdf:about="http://www.semanticweb.org/ulfaamalia/ontologies/2019/9/untitled-ontology-5#Layering">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ulfaamalia/ontologies/2019/9/untitled-ontology-9#NetworkDesignPrinciples"/>
  <owl:disjointWith rdf:resource="http://www.semanticweb.org/ulfaamalia/ontologies/2019/9/untitled-ontology-9#NamingAndAddressing"/>
  <rdfs:label xml:lang="en">Layering</rdfs:label>
</owl:Class>

```

Gambar 3. 4 Penerapan *subClassOf* pada Ontologi

e. Mendefinisikan *facet* (batasan) pada slot: Batasan atau *facet* menggambarkan tipe nilai yang dapat berupa kardinalitas. *Facet* yang umum digunakan menurut (sumber: skripsi ui), diantaranya:

1. *Slot cardinality*: mendefinisikan sejumlah nilai yang dimiliki *slot*. Ada dua jenis, yaitu *single cardinality* (memiliki satu nilai) dan *multiple cardinality* (memiliki beberapa nilai).
2. *Slot value type*: mendefinisikan tipe dari nilai yang dapat mengisi *slot*. Berikut tipe nilai yang umum digunakan diantaranya:
 - String, tipe nilai yang paling umum dan paling sering digunakan untuk *slot*.
 - Number, seperti Float dan Integer yang digunakan untuk menggambarkan *slot* dengan tipe angka atau numerik.
 - Boolean, dimana tipe ini memiliki *yes-no flag*.
 - Enumerated, dimana tipe ini merinci daftar dari spesifik nilai yang diperbolehkan untuk *slot*.

Ontologi yang dikembangkan ini menggunakan tipe *slot value type* dengan tipe *String*.

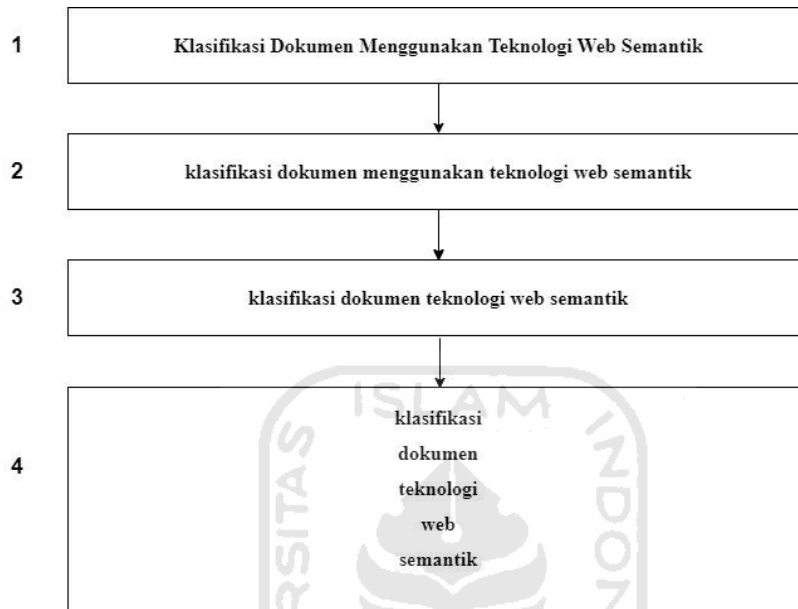
f. Mendefinisikan *instance* (individu): Individu merupakan sesuatu yang direpresentasikan oleh konsep. Tahapan ini sebenarnya bukan tahap wajib karena ontologi sudah merupakan konsep dari domain itu sendiri. Individu sendiri akan digantikan dengan keberadaan *subclass-subclass* yang fungsinya sama seperti individu.

3.3 Tahapan Klasifikasi

Tahapan klasifikasi pada judul tugas akhir melewati beberapa proses, seperti pra pemrosesan teks, penemuan frasa menggunakan proses permutasi, dan mengklasifikasikan frasa yang telah ditemukan sesuai dengan ontologi yang ada.

3.3.1 Pra pemrosesan Teks (*Text Preprocessing*)

Tahapan ini diperlukan untuk mengeliminasi kata-kata yang tidak penting di dalam suatu kalimat agar pengklasifikasian dapat berjalan optimal. Dalam kasus ini, digunakan sebuah judul dokumen tugas akhir mahasiswa untuk mengidentifikasinya. Gambar 3. 5 merupakan contoh judul yang dilakukan pra pemrosesan teks:



Gambar 3. 5 Contoh Penerapan Pra Pemrosesan Teks

Penjelasan proses:

1. Judul sebenarnya (sebelum dilakukan pra pemrosesan teks).
2. *Case folding*: Huruf di dalam judul dibuat seragam. Dalam hal ini, semua huruf diubah menjadi kecil (*lower case*).
3. *Stopword removal*: Menghilangkan kata-kata yang kurang penting.
4. *Tokenization*: Membuat daftar kata (token) dari langkah sebelumnya yang dikenal dengan proses tokenisasi.

Pada proses *stopword removal*, kata yang dihilangkan adalah kata-kata yang kurang penting, mayoritas berupa kata sambung, yang jika dihilangkan tidak akan mempengaruhi inti judul. Seperti contoh pada Gambar 3. 5, kata “menggunakan” termasuk kurang penting dan harus dihilangkan.

3.3.2 Penemuan Kombinasi Kata (Frasa) Menggunakan Permutasi

Penggunaan permutasi setelah kalimat judul telah melewati proses terakhir pra pemrosesan, yaitu tokenisasi, menggunakan rumus P_k^n , dengan $k = 2$. Rumus permutasi secara umum dapat dilihat pada persamaan (2.1).

$$P_k^n = \frac{n!}{(n-k)!}, n > k \quad (2.1)$$

Contoh penerapan:

1. Terdapat lima token atau elemen ($n = 5$) setelah dilakukan pra prosesi: **klasifikasi, dokumen, teknologi, web, semantik.**
2. Token yang ada akan disusun dengan dua token atau elemen ($k = 2$). Jadi, rumus yang digunakan adalah P_2^5 .
3. Hasil perhitungan permutasi P_2^5 :

$$P_2^5 = \frac{n!}{(n-k)!} = \frac{5!}{(5-2)!} = \frac{5.4.3.2.1}{3.2.1} = 20 \quad (2.2)$$

Jadi, terdapat 20 susunan kata (20 frasa) dari lima token atau elemen menggunakan perhitungan permutasi tersebut.

4. Hasil akhir permutasi dari susunan dua token menghasilkan 20 susunan kata (frasa) dengan penjabaran sebagai berikut:
 - a. klasifikasi dokumen
 - b. klasifikasi teknologi
 - c. klasifikasi web
 - d. klasifikasi semantik
 - e. dokumen klasifikasi
 - f. dokumen teknologi
 - g. dokumen web
 - h. dokumen semantik
 - i. teknologi klasifikasi

- j. teknologi dokumen
- k. teknologi web
- l. teknologi semantik
- m. web klasifikasi
- n. web dokumen
- o. web teknologi
- p. web semantik
- q. semantik klasifikasi
- r. semantik dokumen
- s. semantik teknologi
- t. semantik web

Setelah didapatkan beberapa frasa, maka proses selanjutnya adalah pemrosesan kueri SPARQL di dalam ontologi. Frasa tersebut diproses dengan metode *looping* atau perulangan di dalam proses kueri SPARQL.

3.3.3 Kueri SPARQL

Bahasa kueri SPARQL adalah kueri yang biasa digunakan untuk mengambil data dari ontologi. Penggunaan kueri SPARQL bertujuan untuk menghasilkan keluaran kelas konsentrasi di Informatika UII yang sesuai dengan frasa masukan. Karena frasa yang dimasukkan lebih dari satu, frasa tersebut menghasilkan banyak kemungkinan keluaran konsentrasi. Maka dari itu, perlu dilakukan perhitungan untuk keluaran konsentrasi agar mempermudah penggolongan konsentrasi teks judul tersebut.

3.4 Pengujian

Pada tahap pengujian dilakukan dengan memasukkan masukan berupa kalimat judul proposal tugas akhir. Keluaran yang dihasilkan untuk tiap masukan akan dievaluasi untuk mengetahui apakah masukan dapat diproses oleh kueri atau tidak dan apakah menghasilkan keluaran konsentrasi yang sesuai dengan data asli atau tidak. Langkah selanjutnya adalah penghitungan akurasi menggunakan *Multiple Class Confusion Matrix*. *Multiple Class Confusion Matrix* merupakan penghitungan akurasi dengan menggunakan *confusion matrix* seperti pada umumnya dengan tambahan pendefinisian kelas-kelas yang ada sehingga lebih mudah untuk diketahui penyebaran kelasnya setelah dilakukan proses klasifikasi. Contoh *Multiple Class Confusion Matrix* seperti pada Gambar 3. 6.

| | Actual Dog | Actual Cat | Actual Rabbit |
|-------------------|------------|------------|---------------|
| Classified Dog | 23 | 12 | 7 |
| Classified Cat | 11 | 29 | 13 |
| Classified Rabbit | 4 | 10 | 24 |

Gambar 3. 6 Contoh *Multiple Class Confusion Matrix*

Sumber: machinelearningknowledge.ai

Berdasarkan Gambar 3. 6, diagonal kotak hijau menunjukkan nilai *True Positive* (TP) yang dimana sebanyak 23 gambar anjing diklasifikasikan benar sebagai gambar anjing, 29 gambar kucing diklasifikasikan benar sebagai gambar kucing, dan 24 gambar kelinci diklasifikasikan benar sebagai gambar kelinci. Sebaliknya, kotak merah merupakan hasil klasifikasi yang kurang tepat. Sebagai contoh, terdapat 12 gambar kucing yang terklasifikasi sebagai gambar anjing. Dapat disimpulkan bahwa penggunaan *Multiple Class Confusion Matrix* dapat mempermudah pembacaan hasil klasifikasi dari sistem, apakah sesuai dengan kelasnya atau tidak.

Penghitungan akurasi keseluruhan dihitung dengan cara membagi jumlah TP dengan jumlah total keseluruhan yang diklasifikasikan. Rumus akurasi keseluruhan (*overall accuracy*):

$$\text{Overall accuracy} = \frac{TP}{\text{Total number of values}} \times 100\% \quad (3.3)$$

BAB IV

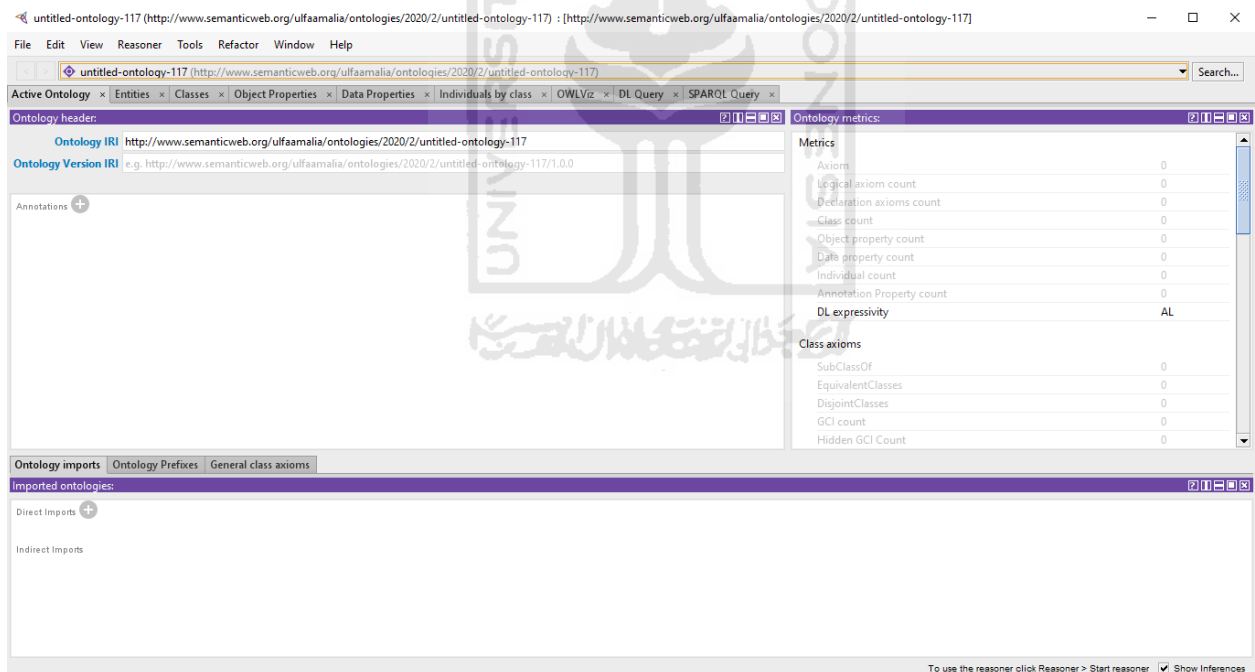
HASIL DAN PEMBAHASAN

4.1 Implementasi Ontologi

Implementasi ontologi menggunakan perangkat lunak Protege versi 5.2. Logo dan tampilan awal Protege ditunjukkan pada Gambar 4. 1 dan Gambar 4. 2.



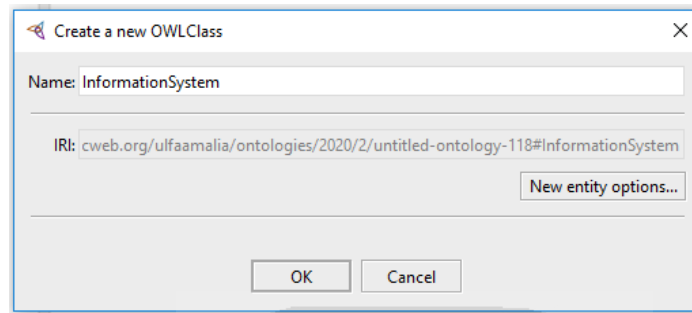
Gambar 4. 1 Logo Protege



Gambar 4. 2 Tampilan awal Protege

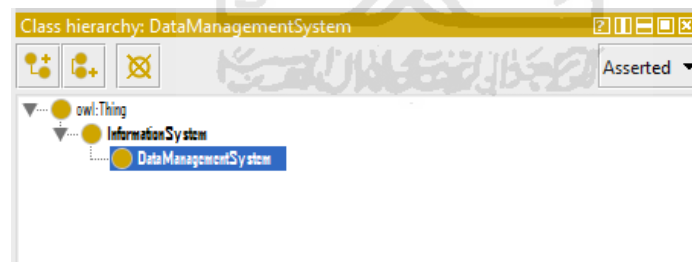
Pada Gambar 4. 2 terdapat beberapa *tab* menu yang memfasilitasi pengembangan ontologi. Pengembangan ontologi dapat dimulai dari membuat kelas di *tab Class*. Pada *superclass Thing* ditambahkan kelas baru menggunakan tombol *Add Subclass*. Subkelas yang dibuat merupakan konsentrasi-konsentrasi yang ada di Informatika UII. Mengenai aturan

penulisan, tidak ada aturan khusus. Namun, alangkah lebih baik jika diawali dengan huruf kapital atau menggunakan aturan *camel case* karena aturan penulisan subkelas pada ontologi tidak memfasilitasi spasi. Bisa juga dengan menggunakan perantara *underscore* (tanda `_`) untuk memisahkan satu kata dengan yang lain. Salah satu contohnya adalah **InformationSystem** yang merepresentasikan **Information System** seperti yang tercantum pada Gambar 4. 3.



Gambar 4. 3 Membuat subkelas dari *Superclass Thing*

Setelah mendefinisikan konsentrasi atau subkelas pada *superclass*, selanjutnya adalah mendefinisikan subkelas dari subkelas, karena tiap konsentrasi mempunyai beberapa cabang, maka diharuskan untuk membuat subkelas lagi di dalamnya. Contoh pada subkelas **InformationSystem** yang memiliki cabang atau subkelas **DataManagementSystem** seperti pada Gambar 4. 4.



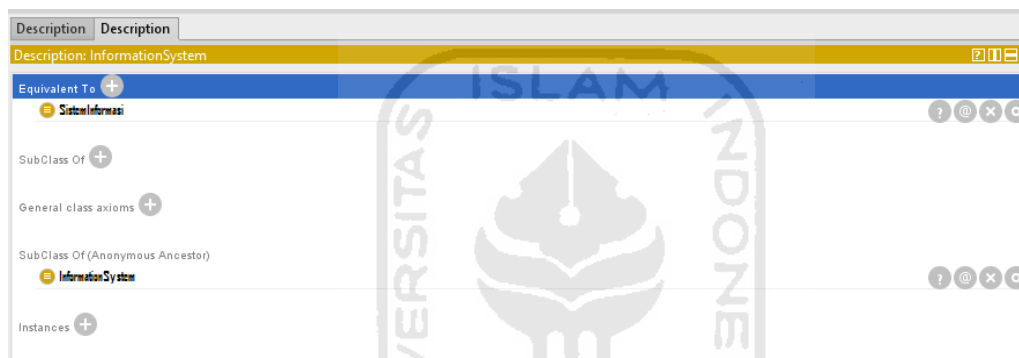
Gambar 4. 4 Contoh subkelas baru pada **InformationSystem**

Penelitian ini dikembangkan untuk klasifikasi judul teks tugas akhir, khususnya menggunakan bahasa Indonesia. Dikarenakan ontologi dari ACM (merujuk pada Bab III) menggunakan bahasa Inggris, maka diperlukan untuk membuat versi bahasa Indonesianya. Caranya dengan membuat subkelas baru di bawah *superclass Thing*. Seperti contoh pada Gambar 4. 5, konsentrasi **Information System** memiliki versi Indonesia **Sistem Informasi** yang ditulis pada ontologi menjadi **SistemInformasi**.



Gambar 4. 5 Subkelas versi Indonesia

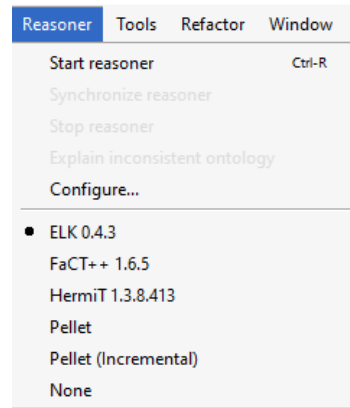
Setelah dibuat versi bahasa Indonesia, kedua kelas (**InformationSystem** dan **SistemInformasi**) perlu dibuat setara. Protege memfasilitasi fitur ini dengan menambahkan salah satu kelas ke deskripsi *Equivalent To* ke kelas yang ingin dibuat setara, seperti yang tercantum pada Gambar 4. 6.



Gambar 4. 6 Deskripsi *Equivalent To* pada subkelas **InformationSystem** terhadap **SistemInformasi**

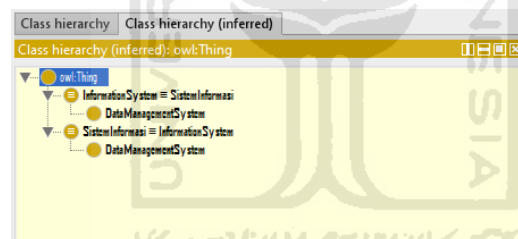
Berdasarkan contoh di atas, subkelas **InformationSystem** memiliki subkelas bernama **DataManagementSystem**. Secara logika, itu artinya subkelas **SistemInformasi** harusnya memiliki subkelas yang sama seperti **InformationSystem**. Langkah ini dapat dilalui dengan melakukan *reasoning*. Proses *reasoning* bertujuan untuk membuat ontologi konsisten dan menyimpulkan konsekuensi logis atau mendapatkan fakta baru dari serangkaian fakta (ontologi) yang dibentuk (Abburu, 2012). Ontologi yang konsisten berperan penting dalam representasi semantik. Ontologi yang tidak konsisten berarti ada kesalahan yang mengakibatkan beberapa konsep dalam ontologi tidak dapat diartikan dengan benar. Ketidakkonsistenan akan menghasilkan pemahaman semantik yang salah (Abburu, 2012).

Dalam Protege, fitur ini sudah tersedia sehingga pengguna dapat dengan mudah melakukan *reasoning* seperti pada Gambar 4. 7.



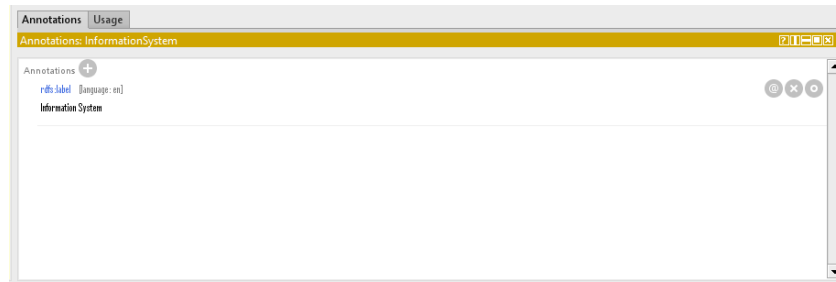
Gambar 4. 7 Reasoner pada Protege

Menurut (sumber: A Survey on Ontology Reasoners and Comparison), beberapa *reasoner* yang populer diantaranya adalah ELK, FaCT++, HermiT, Pellet, CEL, RACER, SWRL-IQ, dan TrOWL. Dalam Protege 5.2, *reasoner* yang tersedia seperti pada Gambar 4.7 yang tercantum di atas. Dalam penelitian ini, penulis menggunakan *reasoner* ELK 0.4.3. Hasil *reasoning* muncul pada tab “**Class hierarchy (inferred)**” yang ditunjukkan pada Gambar 4. 8.



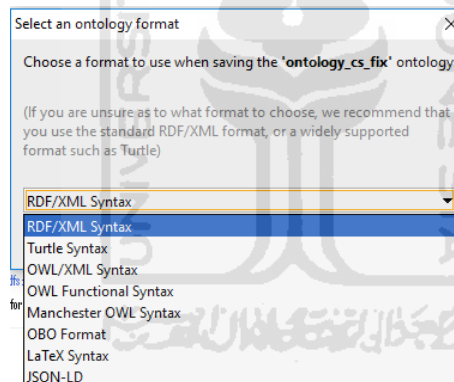
Gambar 4. 8 Ontologi yang setelah di-*reasoning*

Selanjutnya adalah pemberian label. Langkah ini opsional karena pemberian label digunakan agar dapat dibaca dengan mudah oleh pengguna, bukan komputer. Pemberian label nantinya di dalam dokumen ontologi berupa **rdfs:label**. Penulisan label berbeda dengan penulisan pada subkelas karena pada label kita bisa menambahkan spasi atau karakter lainnya agar jelas dibaca oleh pengguna seperti pada **Error! Reference source not found.**



Gambar 4. 9 Pemberian label pada subkelas

Ontologi yang telah dibuat di Protege dapat disimpan dalam berbagai format yang nantinya dapat diakses untuk proses klasifikasi teks judul. Semua tipe dokumen yang disimpan berekstensi OWL, akan tetapi untuk format dokumen bisa disimpan dengan RDF/XML maupun *Turtle Syntax*. Perbedaan format dokumen yang mencolok RDF/XML dan *Turtle Syntax* adalah RDF/XML berupa dokumen XML (karena namanya RDF/XML), sedangkan *Turtle Syntax* berupa dokumen yang berisi *triple*. Perbandingan format penyimpanan dapat ditunjukkan pada Gambar 4. 11 dan Gambar 4. 12.



Gambar 4. 10 Format penyimpanan ontologi dalam Protege

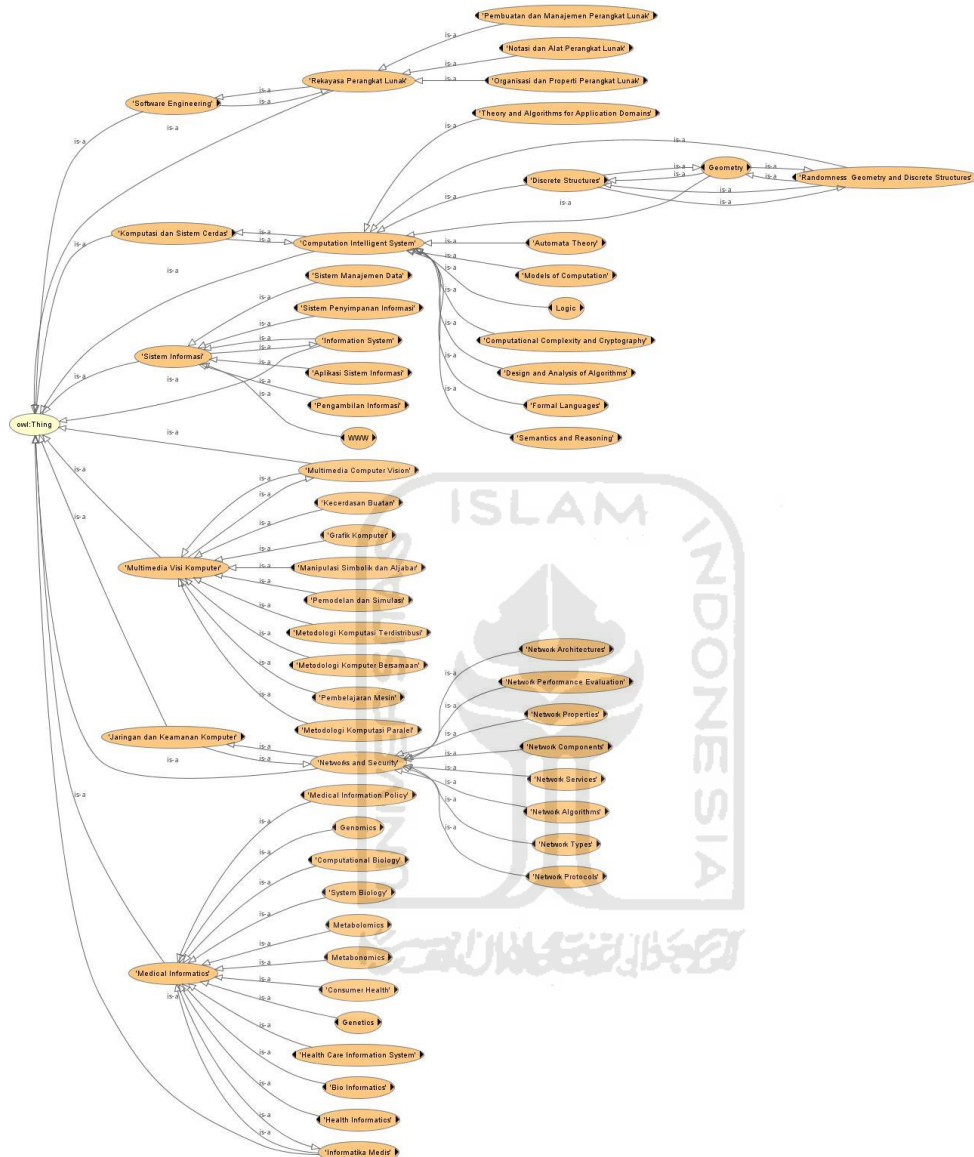
```
<owl:Class rdf:about="http://www.semanticweb.org/ulfaamalia/ontologies/2019/9/untitled-ontology-9#APILanguages">
  <owl:equivalentClass rdf:resource="http://www.semanticweb.org/ulfaamalia/ontologies/2019/9/untitled-ontology-9#BahasaAPI"/>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ulfaamalia/ontologies/2019/9/untitled-ontology-9#ContextSpecificLanguages"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">API Languages</rdfs:label>
</owl:Class>
```

Gambar 4. 11 Format penyimpanan dalam RDF/XML

```
### http://www.semanticweb.org/ulfaamalia/ontologies/2019/9/untitled-ontology-5#Layering
untitled-ontology-5:Layering rdf:type owl:Class ;
  rdfs:subClassOf untitled-ontology-9:NetworkDesignPrinciples ;
  owl:disjointWith untitled-ontology-9:NamingAndAddressing ;
  rdfs:label "Layering"@en .
```

Gambar 4. 12 Format penyimpanan dalam *Turtle Syntax*

Ontologi yang telah dibuat dapat divisualisasikan dengan memanfaatkan *plugin* yang terdapat pada Protege yang bernama OWLViz. Hasil sebagian ontologi yang telah dikembangkan seperti pada Gambar 4. 13.



Gambar 4. 13 Visualisasi ontologi

4.2 Implementasi Klasifikasi

Setelah melalui tahap pengembangan ontologi, tahap selanjutnya adalah implementasi untuk melakukan klasifikasi teks judul. Tahap ini terdiri dari tiga bagian, yaitu pra pemrosesan teks judul, permutasi untuk pembentukan frasa, dan kueri dengan SPARQL. Implementasi pada tahap ini menggunakan IDE Eclipse for Java Developers-2019-12 yang kemudian dilanjutkan di IDE Eclipse JEE, bahasa pemrograman Java serta *library* Apache Lucene dan Jena. Bahasa pemrograman Java dipilih karena prosesnya yang cepat dan banyak penelitian sebelumnya

yang menggunakan Java sehingga dalam implementasinya cukup mudah untuk mencari referensinya.

4.2.1 Pra pemrosesan Teks Judul

Teks judul yang didapat melalui proses pra pemrosesan terlebih dahulu sebelum memasuki tahap permutasi. Proses ini dapat dilihat pada Gambar 4. 14.

```

public class Preprocessing {

    public static StringTokenizer generateTokens(String kalimat)
    throws IOException {

        List<String>stopwords =
Files.readAllLines(Paths.get("res/stopwords_id.txt"));
        String allWords[] = kalimat.toLowerCase().split(" ");
        StringBuilder builder = new StringBuilder();
        for(String judul : allWords) {
            if(!stopwords.contains(judul)) {
                builder.append(judul);
                builder.append(' ');
            }
        }

        String res =
builder.toString().trim().replaceAll("[+.^:, '()]", "");
        System.out.println(res);

        // tokenisasi
        StringTokenizer tokens = new StringTokenizer(res);
        return tokens;
    }

}

```

Gambar 4. 14 Kode program pra pemrosesan teks judul dengan bahasa pemrograman Java

Berdasarkan pada Gambar 4. 14, setelah judul dimasukkan, maka judul akan melalui proses *case folding* yang dimana semua huruf pada judul tersebut diseragamkan menjadi huruf kecil. Setelah semua huruf diseragamkan, maka proses ini berlanjut pada menghilangkan kata-kata yang kurang penting (*stopword removal*) dengan daftar *stopword* terdapat di `stopwords_id.txt`. Proses *case folding* pada tahap sebelumnya berguna pada proses ini karena daftar *stopword* yang penulis miliki terdiri dari huruf-huruf kecil. Jadi, akan lebih mudah diseragamkan terlebih dahulu semua hurufnya dan baru dilakukan proses penghilangan *stopword*. Teks tersebut juga dihilangkan karakter-karakter lain selain huruf dengan bantuan

method `replaceAll()`. Setelah teks judul bersih dari kata-kata dan karakter-karakter yang kurang penting, teks judul tersebut melewati proses tokenisasi (mengubah teks menjadi token). Proses ini melibatkan *library* `StringTokenizer` yang terdapat pada Java.

4.2.2 Permutasi

Proses permutasi bertujuan untuk mendapatkan frasa yang didapatkan dari proses tokenisasi. Permutasi yang digunakan adalah P_2^n , dimana n merupakan jumlah token, sementara 2 adalah jumlah kata yang akan dijadikan frasa. Proses permutasi dipilih daripada kombinasi dikarenakan permutasi menghasilkan lebih banyak kemungkinan frasa daripada kombinasi. Proses ini dapat dilihat pada Gambar 4. 15.

```

public class Permutation {
    List<String> tokens;

    Permutation(List<String> tokens){
        this.tokens = tokens;
    }

    public List<String> generatePermutations() {
        List<String> permuts = new ArrayList<String>();
        int i = 0;

        while(i < tokens.size()) {
            int j = 0;
            while (j < tokens.size()) {
                if (i != j) {
                    System.out.println(this.tokens.get(i)+" "+ this.tokens.get(j));
                    permuts.add(this.tokens.get(i)+" "+ this.tokens.get(j));
                }
                j++;
            }
            i++;
        }
        return permuts;
    }
}

```

Gambar 4. 15 Kode program proses permutasi dengan bahasa pemrograman Java

4.2.3 Implementasi SPARQL

Proses selanjutnya yaitu implementasi ontologi pada bahasa pemrograman Java. Pemrosesan Ontologi pada bahasa pemrograman Java dibantu oleh Apache Jena, yaitu *framework* terbuka untuk web semantik. Apache Jena menyediakan API untuk mengekstrak data dari serta menulis ke model data RDF. Disamping Apache Jena, dibutuhkan juga Apache Lucene, yaitu sebuah *library* yang digunakan untuk pencarian informasi di bidang teks dalam dokumen. Kedua logo *library* tersebut ditunjukkan pada Gambar 4. 16 dan Gambar 4. 17.



Gambar 4. 16 Logo Apache Jena

Sumber: wikipedia.org/wiki/Apache_Jena



Gambar 4. 17 Logo Apache Lucene

Sumber: wikipedia.org/wiki/Apache_Lucene

Tahap sebelum memasuki tahap pemrosesan kueri adalah proses *reasoning* ontologi. Seperti yang dijelaskan pada tahap pengembangan ontologi pada subbab 4.1, proses *reasoning* bertujuan untuk membuat ontologi konsisten dan mendapatkan fakta baru dari ontologi yang dikembangkan. Proses *reasoning* melibatkan *library* `org.apache.jena.ontology.OntModelSpec`.

```

static String URI =
"http://www.semanticweb.org/ulfaamalia/ontologies/2019/9/ontology_cs_fix#
";

public static void main (String [] args) throws IOException {

    String onto = ("res/ontology_cs_new5.owl");

    OntModel model =
ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM_MICRO_RULE_INF);

    model.read(onto, "RDF/XML");

```

Gambar 4. 18 Kode program proses *reasoning* pada ontologi

Pada Gambar 4. 18, URI ontologi didefinisikan pada variabel `URI`. Kemudian, dokumen ontologi juga disimpan pada variabel `onto`. Pada baris kode `OntModel model = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM_MICRO_RULE_INF)` menjelaskan pembentukan model ontologi dengan menggunakan *reasoner* dengan *library* `OntModel` dan `ModelFactory` dengan *field* `CreateOntologyModel`. Proses *reasoning* dapat bekerja dengan memanfaatkan *library* `OntModelSpec` dan *field* `OWL_MEM_MICRO_RULE_INF` yang dimana menjelaskan spesifikasi untuk model OWL yang disimpan dalam memori dan menggunakan mesin inferensi (*reasoning*) aturan OWL mikro untuk mendapatkan fakta atau kesimpulan baru.

Proses selanjutnya adalah implementasi kueri SPARQL. Proses kueri ini bertujuan untuk menghasilkan kelas-kelas konsentrasi dan jumlah kelas yang muncul berdasarkan frasa-frasa yang dimasukkan. Prefix dideklarasikan terlebih dahulu sebagai awalan *namespace* agar tidak perlu menuliskan URI secara berulang (*looping*) di dalam kueri. Berdasarkan pada Gambar 4. 19, variabel `?e` menunjukkan variabel kelas konsentrasi, sedangkan `?countKonsentrasi` menunjukkan variabel jumlah kelas yang muncul (nilai kelas). Setiap frasa yang menjadi masukan dianggap dengan variabel `?x` yang merupakan subkelas dari `b` (variabel `?b`). Variabel `?b` merupakan subkelas dari variabel `?c` dan `?c` merupakan subkelas dari `Thing`,

yang dimana *Thing* adalah *parent class* atau *superclass*. Sebelumnya, masukan frasa perlu diberi ketentuan, yang dimana frasa yang dimasukkan berupa *regular expression (regex)* dan menggunakan tanda `'i'` untuk masukan frasa tidak memperhatikan huruf besar atau kecil (*case-insensitive*). Lain halnya dengan tanda `'%s'` yang merupakan *string format* sebagai tempat masukan frasa yang berulang. Hasil dari proses kueri diurutkan berdasarkan `?countKonsentrasi` dari yang paling besar hingga kecil.

```
String queryString = "PREFIX cs: <" + URI + "> "
                    + "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> "
                    + "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-
ns#> "
                    + "PREFIX owl: <http://www.w3.org/2002/07/owl#> "
                    + "SELECT ?e (COUNT (?e) AS ?countKonsentrasi) "
                    + "WHERE { ?x rdfs:label ?a;"
                    + "      rdfs:subClassOf ?b ."
                    + "      ?b rdfs:subClassOf ?c ."
                    + "      ?c rdfs:subClassOf owl:Thing ."
                    + "      ?c rdfs:label ?e ."
                    + "      FILTER (regex(str(?a), '%s', 'i')) ."
                    + "}"
                    + "GROUP BY ?e "
                    + "ORDER BY DESC (?countKonsentrasi) ";
```

Gambar 4. 19 Kode program kueri SPARQL

Setelah implementasi kueri SPARQL, selanjutnya adalah proses eksekusi kueri SPARQL yang disimpan pada variabel `NewTokens` yang ditunjukkan pada Gambar 4. 20. Variabel tersebut menggunakan *method* `format()` yang mengisi template *string* pada variabel `queryString` (yang berisi kueri SPARQL) dengan daftar frasa yang tersimpan pada variabel `permut` dengan memanfaatkan `'%s'` tadi. Selanjutnya, proses eksekusi ditangani oleh *library* `QueryExecution` dan `QueryExecutionFactory` untuk mengeksekusi `NewTokens` dengan data dari model yang tersimpan pada `qexec`. Pendefinisian `results` berguna untuk menyimpan hasil pengekseskuan kueri `SELECT` (menggunakan *method* `execSelect()`) pada `qexec`.

Karena kueri dilakukan berulang sebanyak jumlah frasa, maka hasil kueri konsentrasi dan nilainya dikeluarkan satu persatu. Hal ini bertujuan untuk mempermudah penggabungan hasil kueri per frasa sebagai hasil akhir sehingga dapat ditarik suatu kesimpulan. Proses ini menggunakan *library* `QuerySolution` yang memunculkan satu hasil keluaran kueri dengan *method* `nextSolution()` agar hasil kueri selanjutnya dapat muncul. Selanjutnya, terdapat

library RDFNode yang memeriksa apakah suatu *node* merupakan *resource* literal atau tidak. Hasil yang ditunjukkan akan seperti ini pada variabel `countKonsentrasi`: `8^http://www.w3.org/2001/XMLSchema#integer`. Hasil literal perlu dibersihkan menggunakan *method* `indexOf()` agar dapat menghasilkan nilai integer saja.

Sesuai tujuan akhir dari penelitian ini yang merujuk pada konsentrasi dan nilai konsentrasi tersebut, maka RDFNode hanya mengembalikan variabel `e` dan `countKonsentrasi` dari kueri SPARQL. Hasil akhirnya disimpan dalam hasil dalam bentuk `ArrayList resultMap`. Implementasi kode dapat ditunjukkan pada Gambar 4. 20.

```
List<Hasil> resultMap = new ArrayList();

for(int i=0 ; i<permut.size(); i++){
    String newTokens = String.format(queryString,
    permut.get(i));
    System.out.println("Kueri frasa: " + newTokens);
    QueryExecution qexec =
    QueryExecutionFactory.create(newTokens,model);

    ResultSet results = qexec.execSelect();
    while (results.hasNext())
    {
        QuerySolution qs = results.nextSolution();
        RDFNode a = qs.get("e");
        RDFNode b = qs.get("countKonsentrasi");
        String konsentrasi = a.toString();
        String count = b.asLiteral().toString();
        int stopperIndex = count.indexOf("h") - 2;
        count = count.substring(0, stopperIndex);
        Hasil hasil = new Hasil(konsentrasi,
        Integer.parseInt(count));
        resultMap.add(hasil);
    }
    qexec.close();
```

Gambar 4. 20 Kode program eksekusi kueri SPARQL

Setelah melewati tahap eksekusi kueri, hasil kueri tersebut perlu digabung dan diurutkan berdasarkan nilai `?countKonsentrasi` dari yang terbesar ke kecil. Pada kesimpulan akhir, konsentrasi dipilih yang memiliki nilai paling besar, yang dimana konsentrasi tersebut berada di indeks ke-0. Proses tersebut ditunjukkan pada Gambar 4. 21. Proses pencarian konsentrasi akan menunjukkan hasil akhir keluaran yang ditunjukkan pada Gambar 4. 22. Begitu pula dengan versi *Graphical User Interface* (GUI), pengguna dapat memasukkan judul tugas akhir, seperti yang tercantum pada Gambar 4. 23, yang kemudian akan menghasilkan peringkat kelas-kelas yang terdapat pada ontologi dan kesimpulan akhir seperti yang ditunjukkan pada Gambar 4. 24.

```

for(int i=0; i<resultMap.size(); i++) {
    System.out.println("Kelas Konsentrasi: "+
resultMap.get(i).getKons() + " -> " + resultMap.get(i).getCount());
}

    System.out.println("\n \nKesimpulan: Judul '" + kalimat +"' termasuk
ke dalam konsentrasi " + resultMap.get(0).getKons()+" atau
"+resultMap.get(1).getKons());

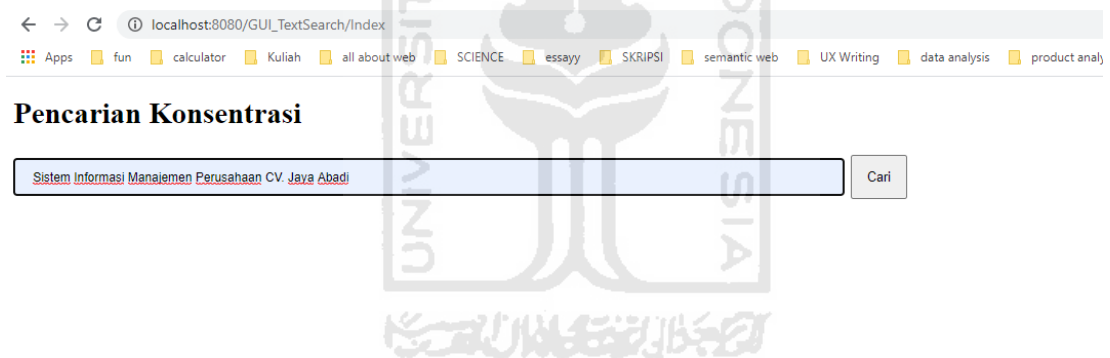
```

Gambar 4. 21 Kode program untuk menampilkan kesimpulan akhir

Kesimpulan: Judul 'Sistem Informasi Manajemen Perusahaan CV. Jaya Abadi' termasuk ke dalam konsentrasi Information System atau Sistem Informasi

Kueri selesai setelah 5131 ms

Gambar 4. 22 Hasil akhir keluaran eksekusi kueri SPARQL setelah diurutkan berdasarkan nilai countKonsentrasi



Gambar 4. 23 Tampilan *search bar* pada GUI Program Pencarian Konsentrasi

Multimedia Information Systems - 2
 Sistem Informasi Enterprise - 2
 Sistem Informasi Geografis - 2
 Sistem Informasi Kesehatan - 2
 Sistem Informasi Multimedia - 2
 Database Management System Engines - 2
 Mesin Sistem Manajemen Basis Data - 2
 learning management system - 2
 sistem manajemen pembelajaran - 2
 File System Management - 2
 Manajemen Sistem File - 2

Kesimpulan: Judul "Sistem Informasi Manajemen Perusahaan CV. Jaya Abadi" termasuk Information System atau Sistem Informasi

Gambar 4. 24 Sebagian tampilan GUI pada hasil Program Pencarian Konsentrasi

Berdasarkan Gambar 4. 24, terdapat beberapa kelas dari ontologi yang muncul setelah dilakukannya pencarian. Kelas yang muncul tidak hanya konsentrasi utamanya, akan tetapi juga subkelas dari konsentrasi utama tersebut. Hal ini terjadi karena proses *reasoning* yang memunculkan semua pengetahuan baru, termasuk semua subkelas dengan bahasa yang berbeda yang terdapat di dalam konsentrasi tersebut. Penjelasan lebih lanjut akan dijabarkan pada paragraf selanjutnya.

Detail daftar yang muncul pada hasil klasifikasi judul "Sistem Informasi Manajemen Perusahaan CV. Jaya Abadi" menurut Gambar 4. 24 adalah sebagai berikut:

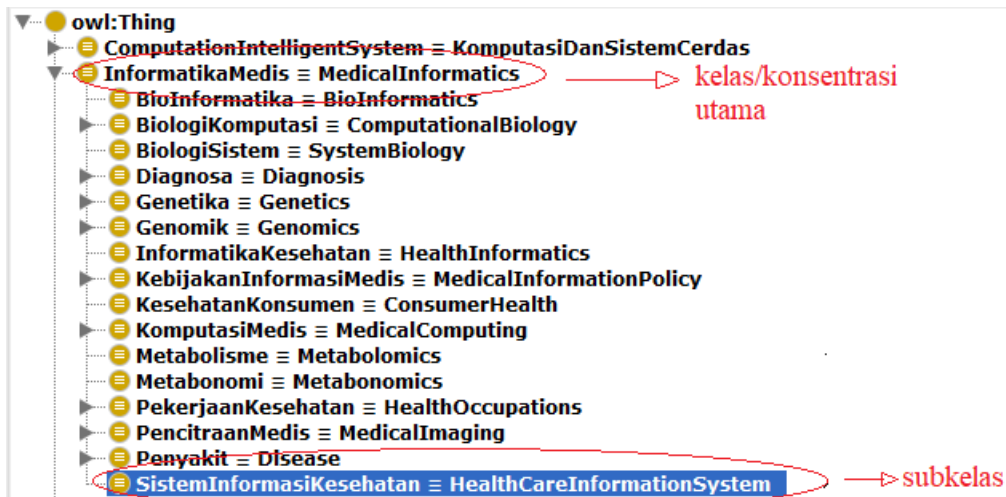
| | |
|--|---|
| Information System - 26 | Software Organization and Properties - 8 |
| Sistem Informasi - 26 | |
| Aplikasi Sistem Informasi - 12 | Data Management System - 6 |
| Information Systems Application - 12 | Sistem Manajemen Data - 6 |
| Information System - 10 | human computer interaction - 6 |
| Sistem Informasi - 10 | interaksi komputer manusia - 6 |
| Rekayasa Perangkat Lunak - 10 | Contextual Software Domains - 6 |
| Software Engineering - 10 | Domain Perangkat Lunak Kontekstual - 6 |
| Multimedia Computer Vision - 8 | Informatika Medis - 4 |
| Multimedia Visi Komputer - 8 | Medical Informatics - 4 |
| Organisasi dan Properti Perangkat Lunak - 8 | Sistem Spasial-temporal - 4 |

| | |
|---|---|
| Spatial-temporal Systems - 4 | Sistem Informasi Geografis - 2 |
| computer aided instruction - 4 | Sistem Informasi Kesehatan - 2 |
| instruksi berbantuan komputer - 4 | Sistem Informasi Multimedia - 2 |
| Operating Systems - 4 | Database Management System Engines - 2 |
| Sistem Operasi - 4 | Mesin Sistem Manajemen Basis Data - 2 |
| Enterprise Information Systems - 2 | learning management system - 2 |
| Geographic Information Systems - 2 | sistem manajemen pembelajaran - 2 |
| Health Care Information System - 2 | File System Management - 2 |
| Multimedia Information Systems - 2 | Manajemen Sistem File - 2 |
| Sistem Informasi Enterprise - 2 | |

Salah satu frasa yang terbentuk adalah “sistem informasi”. Dari satu frasa tersebut menghasilkan beberapa kelas (daftar hasil) dan jumlah kemunculan kelasnya sebagai berikut:

| | |
|--|---|
| Aplikasi Sistem Informasi - 12 | Sistem Spasial-Temporal - 4 |
| Information System - 16 | Spatial-temporal Systems - 4 |
| Information System Application - 12 | Health Care Information System - 2 |
| Sistem Informasi - 16 | Informatika Medis - 4 |
| Enterprise Information Systems - 2 | Medical Informatics - 4 |
| Sistem Informasi Enterprise - 2 | Sistem Informasi Kesehatan - 2 |
| Geographic Information Systems - 2 | Multimedia Information Systems - 2 |
| Sistem Informasi Geografis - 2 | Sistem Informasi Multimedia - 2 |

Karena menggunakan basis kata kunci, kueri akan mencari kelas apa saja yang memiliki unsur “sistem informasi” di dalamnya dan menelusuri hingga ke kelasnya (atau konsentrasi utamanya) sehingga memungkinkan semua konsentrasi dapat muncul selama konsentrasi tersebut memiliki kelas yang memiliki unsur frasa yang dimaksud (penjelasan ada di Gambar 4. 26). Sebagai salah satu contohnya yaitu muncul kelas sekaligus konsentrasi utama “Informatika Medis” dan subkelasnya yaitu “Sistem Informasi Kesehatan” di daftar hasil. Subkelas tersebut sudah jelas terdapat unsur frasa “sistem informasi”. Penjelasan tersebut dapat dilihat pada gambar struktur ontologi pada Gambar 4. 25.



Gambar 4. 25 Gambaran struktur ontologi subkelas dan kelas pada kueri SPARQL

Angka yang muncul di samping kelas tersebut merupakan jumlah kemunculan kelas yang berasal dari masukan frasa. Pada contoh yang sama dan merujuk ke kueri SPARQL pada Gambar 4. 19, frasa ?x berarti “sistem informasi” yang langsung mengarah ke kelas “Sistem Informasi Kesehatan” karena kelas tersebut memiliki unsur frasa yang disebutkan. Hasil *break down* dari penjelasan ini digambarkan pada Gambar 4. 26.

Berdasarkan Gambar 4. 26, baris nomor 1 merujuk ke kelas yang ekuivalen dengan kelas itu sendiri yang merupakan subkelas dari kelas itu sendiri. Baris nomor 2 dan 3 menjelaskan bahwa kelas “Sistem Informasi Kesehatan” masih menjadi ekuivalen dari “Health Care Information System” dan menjadi subkelas dari “Medical Informatics” dan “Informatika Medis”. Sekali lagi, ini dikarenakan efek dari *reasoning* yang mengakibatkan semua kelas yang memiliki bahasa yang berbeda ikut masuk ke dalam subkelas. Sedangkan baris nomor 4 sama seperti baris nomor 1 yang kemudian ia merupakan subkelas dari kelas yang ekuivalen dengan kelas itu sendiri dan seterusnya hingga baris ke-12. Jika menggunakan dua bahasa, maka hasil bahasa Indonesia dan bahasa Inggris akan sama. Lalu, untuk fungsi yang menggabungkan semua hasil kueri menggunakan fungsi pada Gambar 4. 20.

Gambaran dari salah satu masukan frasa tersebut menunjukkan bahwa frasa “sistem informasi” menghasilkan jumlah kemunculan kelas paling banyak terdapat pada “Sistem Informasi” yang ekuivalen dengan “Information System”. Ini belum termasuk frasa yang lain yang dimana hasilnya bisa lebih kompleks dan dapat mendukung bahwa judul tersebut termasuk ke konsentrasi “Sistem Informasi” atau “Information System”.

| *?x dengan label ?a | ?b | ?c | ?e | Jumlah |
|----------------------------------|---|---|----------------------------------|--------|
| 1. "Sistem Informasi Kesehatan" | <untitled-ontology-9#HealthCareInformationSystem> | <untitled-ontology-9#HealthCareInformationSystem> | "Health Care Information System" | 1 |
| 2. "Sistem Informasi Kesehatan" | <untitled-ontology-9#HealthCareInformationSystem> | <untitled-ontology-9#InformatikaMedis> | "Informatika Medis" | 1 |
| 3. "Sistem Informasi Kesehatan" | <untitled-ontology-9#HealthCareInformationSystem> | <untitled-ontology-9#MedicalInformatics> | "Medical Informatics" | 1 |
| 4. "Sistem Informasi Kesehatan" | <untitled-ontology-9#HealthCareInformationSystem> | <untitled-ontology-9#SistemInformasiKesehatan> | "Sistem Informasi Kesehatan" | 1 |
| 5. "Sistem Informasi Kesehatan" | <untitled-ontology-9#InformatikaMedis> | <untitled-ontology-9#InformatikaMedis> | "Informatika Medis" | 1 |
| 6. "Sistem Informasi Kesehatan" | <untitled-ontology-9#InformatikaMedis> | <untitled-ontology-9#MedicalInformatics> | "Medical Informatics" | 1 |
| 7. "Sistem Informasi Kesehatan" | <untitled-ontology-9#MedicalInformatics> | <untitled-ontology-9#InformatikaMedis> | "Informatika Medis" | 1 |
| 8. "Sistem Informasi Kesehatan" | <untitled-ontology-9#MedicalInformatics> | <untitled-ontology-9#MedicalInformatics> | "Medical Informatics" | 1 |
| 9. "Sistem Informasi Kesehatan" | <untitled-ontology-9#SistemInformasiKesehatan> | <untitled-ontology-9#HealthCareInformationSystem> | "Health Care Information System" | 1 |
| 10. "Sistem Informasi Kesehatan" | <untitled-ontology-9#SistemInformasiKesehatan> | <untitled-ontology-9#InformatikaMedis> | "Informatika Medis" | 1 |
| 11. "Sistem Informasi Kesehatan" | <untitled-ontology-9#SistemInformasiKesehatan> | <untitled-ontology-9#MedicalInformatics> | "Medical Informatics" | 1 |
| 12. "Sistem Informasi Kesehatan" | <untitled-ontology-9#SistemInformasiKesehatan> | <untitled-ontology-9#SistemInformasiKesehatan> | "Sistem Informasi Kesehatan" | 1 |

Kesimpulan Jumlah:

Health Care Information System - 2
 Sistem Informasi Kesehatan - 2
 Medical Informatics - 4
 Informatika Medis - 4

*Keterangan: ?x dengan label ?a artinya ?x memiliki nama lain ?a untuk mempermudah cara baca oleh pengguna. Penjelasan ini ada di Bab IV bagian Implementasi Ontologi.

Gambar 4. 26 Salah satu hasil *break down* dari kueri SPARQL pada frasa “sistem informasi”

4.3 Pengujian

Sesuai dengan rencana pengujian pada Bab 3, pengujian dilakukan dengan memasukkan beberapa masukan judul proposal tugas akhir yang diperoleh dari Jurusan Informatika UII tahun 2019. Total judul yang digunakan untuk pengujian sebanyak 60 yang diasumsikan benar dengan pembagian masing-masing konsentrasi sebanyak 10 judul. Kemudian penghitungan akurasi menggunakan *Multiple Class Confusion Matrix* yang ditunjukkan pada Tabel 4. 1. Konsentrasi yang dihitung di sini adalah konsentrasi utama pada bagian “kesimpulan akhir” yang terdapat pada program.

Tabel 4. 1 Hasil pengujian menggunakan *Multiple Class Confusion Matrix* (1)

| <i>Confusion Matrix</i> | | ACTUAL | | | | | | Total |
|---|--------------|-----------|-----------|-----------|-----------|-----------|-----------|--------------|
| | | MVK | IM | JKK | SI | RPL | KSC | |
| P R E D I C T E D | MVK | 2 | 2 | 1 | - | - | - | 5 |
| | IM | - | 3 | - | - | - | - | 3 |
| | JKK | - | - | 4 | - | - | - | 4 |
| | SI | - | 3 | 1 | 7 | 2 | - | 13 |
| | RPL | - | - | - | - | 5 | - | 5 |
| | KSC | 1 | 1 | 1 | - | - | 7 | 10 |
| | None | 7 | 1 | 3 | 3 | 3 | 3 | 20 |
| | Total | 10 | 10 | 10 | 10 | 10 | 10 | 60 |

Keterangan:

MVK : Multimedia Visi Komputer

IM : Informatika Medis

JKK : Jaringan dan Keamanan Komputer

SI : Sistem Informasi

RPL : Rekayasa Perangkat Lunak

KSC : Komputasi dan Sistem Cerdas

None : judul tidak terdeteksi ke konsentrasi apapun

Nilai akurasi keseluruhan (*Overall Accuracy*):

$$\begin{aligned}
 \text{Overall Accuracy} &= \frac{TP}{\text{Total number of values}} \times 100\% & (3.1) \\
 &= \frac{2 + 3 + 4 + 7 + 5 + 7}{60} \times 100\% \\
 &= \frac{28}{60} \times 100\% = 46,67\%
 \end{aligned}$$

Berikutnya adalah tahap pengujian kedua menggunakan *actual data* yang berasal dari pakar (dalam penelitian ini adalah dosen Informatika UII) dengan menambahkan sedikit pengetahuan baru di dalam ontologi. Jumlah data yang digunakan sebanyak 60 data judul TA dengan catatan terdapat lima judul yang terklarifikasi dua konsentrasi oleh pakar (totalnya menjadi 65 judul). Hasil pengujian dapat dilihat pada Tabel 4. 2.

Tabel 4. 2 Hasil pengujian menggunakan *Multiple Class Confusion Matrix* (2)

| <i>Confusion Matrix</i> | | ACTUAL | | | | | | Total |
|---|--------------|-----------|----------|-----------|-----------|----------|-----------|--------------|
| | | MVK | IM | JKK | SI | RPL | KSC | |
| P R E D I C T E D | MVK | 5 | - | - | 1 | 1 | - | 7 |
| | IM | - | 3 | - | - | - | - | 3 |
| | JKK | - | - | 10 | - | - | - | 10 |
| | SI | - | 5 | - | 8 | - | 2 | 15 |
| | RPL | - | - | - | 5 | 7 | 1 | 13 |
| | KSC | 1 | 1 | - | - | - | 7 | 9 |
| | None | 4 | - | 1 | 2 | - | 1 | 8 |
| | Total | 10 | 9 | 11 | 16 | 8 | 11 | 65 |

Nilai akurasi keseluruhan (*Overall Accuracy*):

$$\begin{aligned}
 \text{Overrral Accuracy} &= \frac{TP}{\text{Total number of values}} \times 100\% & (3.2) \\
 &= \frac{5 + 3 + 10 + 8 + 7 + 7}{65} \times 100\% \\
 &= \frac{40}{65} \times 100\% = 61,53\%
 \end{aligned}$$

Berdasarkan dari hasil pengujian pertama yang telah dilakukan pada 60 judul, terdapat 40 judul yang dapat terdeteksi dan 20 judul yang tidak terdeteksi. Dari 40 judul tersebut, terdapat beberapa judul yang tidak terklasifikasi sesuai dengan judul sebenarnya (*actual*). Hasil dari penghitungan akurasi keseluruhan sebesar 46,67%. Sedangkan setelah dilakukannya pengujian yang kedua menggunakan data asli dari pakar dan pengetahuan tambahan di ontologi, nilai akurasinya sebesar 61,53%.

Nilai akurasi di atas menunjukkan bahwa penggunaan ontologi dirasa masih belum cukup baik untuk klasifikasi. Menurut penulis, ada beberapa faktor yang mempengaruhi nilai akurasi ini, seperti data judul yang digunakan merupakan asumsi benar, yang dimana bisa jadi konsentrasi asli (atau data *actual*) di dalam data tersebut tidak sepenuhnya benar. Hal ini dibuktikan dengan kenaikan nilai akurasi sebesar 14,86% karena adanya perubahan data judul yang berasal dari pakar. Selain itu, bisa jadi fitur yang digunakan untuk klasifikasi masih terbatas pada judul sehingga pembentukan frasa juga ikut terbatas. Terakhir, dari segi sumber datanya, yaitu ontologi, yang dikembangkan belum cukup kaya pengetahuannya sehingga belum cukup representatif. Dari hasil pengujian kedua setelah ditambahkannya sedikit pengetahuan baru, terdapat hasil yang cukup signifikan pada pengujian kedua.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian, dapat ditarik beberapa kesimpulan yang mengacu pada tujuan dari penelitian ini. Diantaranya adalah:

- a. Pendekatan semantik dapat membantu untuk proses klasifikasi judul tugas akhir dengan ontologi sebagai teknologi semantik sekaligus representasi data. Ontologi dapat dikembangkan sesuai dengan kebutuhan. Dalam kasus ini, sumber ontologi didapat dari *Association for Computing Machinery* (ACM) dan *Computer Science Ontology* (CSO) serta tambahan ontologi untuk memperkaya konsentrasi Informatika Medis yang bersumber dari *Medical Subject Headings* (MeSH) dan *International Classification of Disease Version 10* (ICD10). Ontologi yang dikembangkan disesuaikan dengan konsentrasi-konsentrasi yang ada di Jurusan Informatika UII yang berpatokan pada Panduan Kurikulum 2010. Selain itu, ontologi dapat dimanfaatkan sebagai kunci klasifikasi teks sehingga dapat dilakukan tanpa *classifier training* atau *training set*. Teks judul tugas akhir akan dipecah jadi beberapa frasa sehingga dapat diketahui konsentrasi apa yang terkandung dalam teks judul tersebut.
- b. Pengujian menggunakan metode *Multiple Class Confusion Matrix* dengan mengimplementasikan data dari pakar menunjukkan hasil akurasi keseluruhan sebesar 61,53% yang dimana penggunaan ontologi untuk klasifikasi judul tugas akhir dinilai belum bekerja dengan baik.

5.2 Saran

Melalui hasil pengujian yang telah dilakukan, terdapat beberapa saran yang dapat diimplementasikan untuk penelitian ke depannya. Diantaranya yaitu:

- a. Perlunya perbaikan pada ontologi untuk menghasilkan ontologi yang lebih kaya pengetahuannya sehingga dapat melakukan proses klasifikasi yang lebih optimal.
- b. Proses klasifikasi yang tidak hanya berdasarkan pada teks judul tugas akhir, akan tetapi konten proposal yang lain yang mewakili judul, seperti bagian abstrak.
- c. Penggunaan frasa sebaiknya lebih diteliti lebih lanjut apakah efektif sebagai acuan kata kunci atau tidak dan perlunya penerapan *machine learning* agar akurasi meningkat.

DAFTAR PUSTAKA

- Abburu, S. (2012). *A Survey on Ontology Reasoners and Comparison*. 57(17), 33–39.
- Admojo, F. T., & Winarko, E. (2016). *Sistem Pencarian Informasi Berbasis Ontologi untuk Jalur Pendakian Gunung Menggunakan Query Bahasa Alami dengan Penyajian Peta Interaktif*. 10(1), 23–34.
- Allahyari, M., Kochut, K. J., & Janik, M. (2014). *Ontology-based Text Classification into Dynamically Defined Topics*.
- Ayedh, A., TAN, G., Alwesabi, K., & Rajeh, H. (2016). The Effect of Preprocessing on Arabic Document Categorization. *Algorithms*, 9(2). <https://doi.org/10.3390/a9020027>
- Beysolow, T. (2018). *Applied Natural Language Processing with Python*. Apress.
- Chaer, A. (1990). *Pengantar Semantik Bahasa Indonesia*. Rineka Cipta.
- Chaer, A. (2007). *Linguistik Umum*. Rineka Cipta.
- Chen, M., Jin, X., & Shen, D. (2011). *Short Text Classification Improved by Learning Multi-Granularity Topics*. 1776–1781.
- Cover, R. (2006). *XML and Web Services In The News*. xml.org/xml/news/archives/archive.10062006.shtml#5
- Durga, A. K., & Govardhan, A. (2011). *Ontology Based Text Categorization - Telugu Documents*. 2(9), 1–4.
- Februariyanti, H. (2012). Klasifikasi Dokumen Berita Teks Bahasa Indonesia menggunakan Ontologi. *Teknologi Informasi DINAMIK*, 17(1), 14–23. <http://www.unisbank.ac.id/ojs/index.php/fti1/article/view/1612/594>
- Fraihat, S. (2015). Ontology-Concepts Weighting For Enhanced Semantic Classification of Documents. *Journal of King Saud University - Computer and Information Sciences*, 27(4), 437–449. <https://doi.org/10.1016/j.jksuci.2015.02.003>
- Hao-jin, T. (2013). Semantic dictionary based method for short text classification. *The Journal of China Universities of Posts and Telecommunications*, 20(August), 15–19. [https://doi.org/10.1016/S1005-8885\(13\)60256-3](https://doi.org/10.1016/S1005-8885(13)60256-3)
- Hastuti, S. (1993). *Pegangan Kuliah Pendidikan Bahasa Indonesia*. UPP IKIP Yogyakarta.
- Horrige, M. (2009). *A Practical Guide to Building OWL Ontologies Using Protege 4 and CO-ODE Tools*. The University of Manchester.
- Jackson, P., & Moulinier, I. (2002). *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization (Natural Language Planning)*. John

Benjamins Publishing Co.

- Jasser, M. M. A. (2015). *Ontology-Based Arabic Documents Classification*.
- Kannan, S., & Gurusamy, V. (2015). *Preprocessing Techniques for Text Mining*. October 2014.
- Kastrati, Z. (2017). *Supervised Ontology-Based Document Classification Model*. 245–251.
- Keraf, G. (1984). *Diksi dan Gaya Bahasa*. Gramedia.
- Krishnaraj, D. N., Kumar, D. P., & Bhagavan, S. K. (2018). Conceptual Semantic Model for Web Document Clustering Using Term Frequency. *EAI Endorsed Transactions on Energy Web*, 5(20), 155744. <https://doi.org/10.4108/eai.12-9-2018.155744>
- Nandini, D. (2014). *Semantic Web And Ontology*. Bookboon.com.
- Noy, N. F., & Mcguinness, D. L. (2000). *Ontology Development 101 : A Guide to Creating Your First Ontology*. 1–25.
- Ontotext. (n.d.). *What is Semantic Technology?*
<https://www.ontotext.com/knowledgehub/fundamentals/semantic-web-technology/>
- P, A. H., & Abdullah, A. (2013). *Linguistik Umum*. Erlangga.
- Peng, X., & Choi, B. (2005). *DOCUMENT CLASSIFICATIONS BASED ON WORD SEMANTIC HIERARCHIES*. 362–367.
- Qazi, A., & Goudar, R. H. (2018). An Ontology-based Term Weighting Technique for Web Document Categorization. *Procedia Computer Science*, 133, 75–81. <https://doi.org/10.1016/j.procs.2018.07.010>
- Rahma, F., Pangestuti, D. D., Herdiani, A., & Selviandro, N. (2019). *Analysis and Implementation of Ontology Based Text Classification on Criminality Digital News*
Analysis and Implementation of Ontology Based Text Classification on Criminality Digital News. <https://doi.org/10.1088/1757-899X/662/2/022135>
- Rosario, E., Rosario, J., Nieva, M., Tan, T., & Tangkeko, M. (2016). *CollaborateIT : A CCS IT Thesis Portal with Electronic Document Management System*. 4, 4–9.
- Rosen, K. H. (2012). *Discrete Mathematics and Its Application*. McGraw-Hill.
- Sabuncu, O., Akpınar, S., Cicekli, N. K., Alpaslan, F. N., & Kara, S. (2012). *An ontology-based retrieval system using semantic indexing*. 37, 294–305. <https://doi.org/10.1016/j.is.2011.09.004>
- Sasangka, S. S. T. W. (2014). *Seri Penyuluhan Bahasa Indonesia: Kalimat*. Kementerian Pendidikan dan Kebudayaan.
- Sembada, A. A. (2018). *Implementasi Klasifikasi dan Pengecekan Kemiripan Judul Tugas*

Akhir. Universitas Islam Indonesia.

Szeredi, P. (2014). *The Semantic Web Explained The Technology and Mathematics behind Web 3.0*. Cambridge University Press.

TopQuadrant Technology Briefing. (2003). *Semantic Technology Version 1.1*. TopQuadrant, Inc.

W3C. (n.d.). *RDF Schema 1.1*. Retrieved March 18, 2020, from <https://www.w3.org/TR/rdf-schema/>

W3C. (2013). *OWL*. <https://www.w3.org/OWL/>

W3C. (2014). *RDF*. <https://www.w3.org/RDF/>

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). *Hierarchical Attention Networks for Document Classification*. 1480–1489.



LAMPIRAN

Lampiran A. Daftar *Stopword*

| Daftar <i>Stopword</i> yang Digunakan | | |
|--|-------------|--------------|
| atau | khususnya | sebuah |
| berdasarkan | kini | sedikit |
| biasa | lain | sekali |
| bagi | lalu | sekaligus |
| banyak | lebih | sekitar |
| beberapa | macam | selain |
| berbasis | menentukan | semua |
| bermacam | menggunakan | semacam |
| dan | melalui | sesudah |
| dapat | oleh | setiap |
| dari | pada | selama |
| daripada | per | tanpa |
| dengan | penyebab | tentang |
| di | secara | tertentu |
| dini | sebagai | terlebih |
| diri | sebagaimana | tiap |
| dulu | sebanyak | terdiri |
| dalam | seberapa | terhadap |
| hal | sebelum | untuk |
| hingga | sehingga | yaitu |
| ini | serupa | yakni |
| karena | sama | yang |
| ke | se | menunjang |
| kemudian | sebab | memanfaatkan |

Lampiran B. Tabel Pengujian 1

| No. | Teks Judul | Kesimpulan Konsentrasi (predicted) | Konsentrasi Sebenarnya (actual) |
|------------|---|---|--|
| 1. | MEDIA PEMBELAJARAN PENGENALAN FLORA DAN FAUNA DALAM BAHASA INGGRIS UNTUK SISWA SEKOLAH DASAR (Studi Kasus Kelas 1,2, dan 3) | MVK | MVK |
| 2. | Gim Edukasi Doa Sehari-hari Untuk Anak TK Melalui Pendekatan Animasi | - | MVK |
| 3. | PENGEMBANGAN SISTEM ANOTASI CITRA MIKROSKOPIK MENGGUNAKAN ANDROID | - | MVK |
| 4. | Media Pembelajaran Aqidah dan Akhlak Untuk Anak Usia Dini Berbasis Multimedia | - | MVK |
| 5. | Pemodelan pemilahan citra gunung Merapi secara otomatis pada BPPTKG Yogyakarta | - | MVK |
| 6. | Peta Digital Situs-Situs Arkeologi di Kabupaten Boyolali | - | MVK |
| 7. | GIM BAHASA INGGRIS DENGAN PENDEKATAN | KSC | MVK |

| | | | |
|-----|--|-----|-----|
| | PEMBELAJARAN KOLABORATIF MENGGUNAKAN METODE MEMORIZING | | |
| 8. | Senam Ibu Hamil Dalam Animasi 3D Berbasis Android | - | MVK |
| 9. | GIM Pencarian Pecahan Bukti Sejarah Kerajaan Mataram Islam Menggunakan Augmented Reality Berbasis Android | - | MVK |
| 10. | Rekonstruksi Tiga Dimensi Kawah Gunung Merapi Berdasarkan Citra Dua Dimensi Kawah Gunung Merapi Secara Otomatis | MVK | MVK |
| | | | |
| 11. | IMPLEMENTASI WEMOS SEBAGAI MIKROKONTROLER BERBASIS INTERNET OF THINGS UNTUK MONITORING KEBAKARAN MELALUI TELEGRAM | JKK | JKK |
| 12. | SISTEM MONITORING KONDISI AIR DI KOLAM RENANG TIRTASARI BERBASIS IOT | - | JKK |
| 13. | Perancangan Proses Bisnis Printer Online Print-in | SI | JKK |

| | | | |
|-----|---|-----|-----|
| | dengan Metode Berbasis Web dan Online Payment | | |
| 14. | Monitoring kondisi air pada kolam pemancingan ikan menggunakan mikrokontroler NodeMCU berbasis internet of things | JKK | JKK |
| 15. | Sistem Pelaporan Dokumen Barang Bukti Berbasis Teknologi Blockchain | KSC | JKK |
| 16. | KONSOLIDASI DAN VISUALISASI LOG SERVER BSI UII MENGGUNAKAN ELK | JKK | JKK |
| 17. | Maltego Scripting Language Untuk Mendukung Tahap Reconnaissance Pada Penetration Test | JKK | JKK |
| 18. | Pengembangan Aplikasi Vulnerability Assessment Berdasarkan OWASP Top 10 Studi Kasus OWASP Top 10 Tahun 2013 Injection, Broken Authentication and Session Management, Cross-Site Scripting, Insecure Direct Object References, Security Misconfiguration (A1 – A5) | JKK | JKK |
| 19. | IOT Untuk Optimasi Pemberian Pakan Ikan | - | JKK |

| | | | |
|-----|--|-----|-----|
| 20. | Penggunaan Multiple Kriptografi dan Steganography Berbasis Android untuk penyembunyian Pesan Teks pada Citra Digital | MVK | JKK |
| | | | |
| 21. | Sistem Diagnosa Penyakit Alzheimer dengan menggunakan Metode Tree | IM | IM |
| 22. | Sistem Monitoring Kesehatan Mental Pada Pegawai | IM | IM |
| 23. | DETEKSI PIGMEN BINTIK HITAM KULIT PADA CITRA WAJAH DENGAN TEKNIK PENGOLAHAN CITRA | MVK | IM |
| 24. | Sistem Deteksi Warna Kulit Tidak Merata Pada Wajah Menggunakan Teknik Pengolahan Citra Berbasis Web | MVK | IM |
| 25. | Sistem Informasi Manajemen Apotek Di Klinik Rakha Farma | SI | IM |
| 26. | SISTEM PENDUKUNG KEPUTUSAN UNTUK DIAGNOSIS DINI GANGGUAN KESEHATAN PADA | KSC | IM |

| | | | |
|-----|---|----|----|
| | HEWAN PELIHARAAN KUCING RAS PERSIA | | |
| 27. | Sistem Pakar Pemilihan Permainan Sebagai Pendukung Perkembangan Anak | SI | IM |
| 28. | Sistem Pendekteksi Bakteri Mycobacterium Tuberculosis dan Penentu Tingkat Penyakit TBC | IM | IM |
| 29. | SISTEM PAKAR UNTUK MENENTUKAN TINGKAT RESIKO PENYAKIT JANTUNG KORONER DENGAN SISTEM INFERENSI FUZZY METODE TSUKAMOTO | SI | IM |
| 30. | PROTOTYPE PENDETEKSI KEMERAHAN DI WAJAH SECARA OTOMATIS MELALUI CITRA WAJAH | | IM |
| 31. | Sistem informasi member parkir dengan perangkat IoT | SI | SI |
| 32. | Sistem Informasi Pemesanan Coworking Space Berbasis Web | SI | SI |
| 33. | SISTEM INFORMASI PRAKTIK DETEKSI HAMBATAN | SI | SI |

| | | | |
|-----|--|----|----|
| | PSIKOLOGIS DAN RANCANGAN PEMBELAJARAN INDIVIDU PADA ANAK BERKESULITAN BELAJAR | | |
| 34. | DESAIN INTERAKSI APLIKASI PEMETAAN MINAT UNTUK ANAK PUTUS SEKOLAH MENGGUNAKAN METODE DESIGN THINKING (STUDI KASUS YAYASAN RUMAH IMPIAN YOGYAKARTA) | - | SI |
| 35. | Sistem Informasi Manajemen Keuangan pada CV. Karya Jati Madiun | SI | SI |
| 36. | Sistem Informasi Pengolahan Data Penduduk Pada Kantor Kepala Desa Rejo Basuki | SI | SI |
| 37. | Sistem Dashboard Monitoring Situasi Akademik di Lingkungan FTI UII | - | SI |
| 38. | Sistem Informasi Manajemen Kapal Kelud PT. PELNI (Persero) | SI | SI |

| | | | |
|-----|--|-----|-----|
| 39. | Sistem informasi penyewaan lapangan futsal di yogyakarta berbasis website | SI | SI |
| 40. | Go Mountain Manajemen Penjadwalan Pelayanan Jasa Porter Gunung Berbasis Web | - | SI |
| | | | |
| 41. | Automated Testing pada Website Marketplace dengan PHPUnit | - | RPL |
| 42. | PEMODELAN DAN PENGUJIAN FUNGSIONALITAS WEBSITE BERBASIS UML ACTIVITY DIAGRAM | RPL | RPL |
| 43. | Unit Testing Pada Aplikasi Web Mobile | RPL | RPL |
| 44. | PENGEMBANGAN APLIKASI BERBASIS WEB DENGAN METODOLOGI BEHAVIOR-DRIVEN DEVELOPMENT (STUDI KASUS: PRINT ONLINE) | RPL | RPL |
| 45. | Sistem Rekomendasi penerimaan anggota baru | SI | RPL |
| 46. | Pengembangan Dashboard Sistem Informasi Rumah Sakit menggunakan metode | SI | RPL |

| | | | |
|-----|--|-----|-----|
| | Data Driven Development Studi Kasus CV Medika Buana Informatika. | | |
| 47. | Desain Interaksi Aplikasi Mobile Waktukita.com menggunakan metode Lean UX | - | RPL |
| 48. | PENGEMBANGAN FRONTEND APLIKASI MOBILE E- RECRUITMENT MENGUNAKAN FLUTTER | RPL | RPL |
| 49. | Peningkatan Kualitas Perangkat Lunak dengan Refactoring | RPL | RPL |
| 50. | APLIKASI CHATBOT UNTUK PENGAJUAN PROSES PEMESANAN SURAT DI FAKULTAS TEKNOLOGI INDUSTRI UNIVERSITAS ISLAM INDONESIA (FTI UII) | | RPL |
| | | | |
| 51. | Membangun Sistem Pendukung Keputusan berbasis Causal Model untuk Menganalisis Kinerja Guru Taman Kanak-Kanak | KSC | KSC |
| 52. | Sistem pendukung keputusan pemilihan tempat wisata yogyakarta dengan | KSC | KSC |

| | | | |
|-----|--|-----|-----|
| | menggunakan metode simple additive weighting (saw) | | |
| 53. | Pemodelan Topik untuk Penelitian di Bidang Informatika | KSC | KSC |
| 54. | Analisis Pengaruh Nilai Ujian Nasional Terhadap Performa Akademik Mahasiswa Menggunakan Metode K-Means | - | KSC |
| 55. | Sistem Pengambilan Keputusan Bantuan Rekomendasi KRS dan Topik Penelitian bagi Mahasiswa menggunakan Metode Tabel Keputusan. | KSC | KSC |
| 56. | Sistem Pendukung Keputusan Penilaian Kinerja Karyawan (Studi Kasus PT. Energi Bumi Sakti Jobsite Tanjung Enim) | KSC | KSC |
| 57. | Analisis Sentimen Berbasis Fitur pada Ulasan Tempat Wisata Menggunakan Metode Convolutional Neural Network (CNN) | KSC | KSC |
| 58. | Deteksi Konten Dewasa pada Tweet Berbahasa Indonesia dengan Metode Long Short-Term Memory (LSTM) Networks | - | KSC |

| | | | |
|-----|--|-----|-----|
| 59. | Analisis Performa Akademik Mahasiswa Teknik Informatika Menggunakan Metode Clustering | - | KSC |
| 60. | Pemodelan Topik terhadap Analisis Sentimen Calon Presiden Republik Indonesia pada Media Sosial Twitter | KSC | KSC |



Lampiran C. Tabel Pengujian 2

| No. | Judul TA | Kesimpulan Konsentrasi (predicted) | Konsentrasi Sebenarnya (actual) |
|-----|---|--|---------------------------------------|
| 1. | Sistem Informasi Geografis Lokasi Toko Oleh-Oleh Khas Yogyakarta Berbasis Android | SI | SI |
| 2. | Sistem Aplikasi Pesan Antar Makanan Berbasis Web Di Omah Pawon | RPL | SI |
| 3. | Perancangan Sistem Informasi Geografis Puskesmas Di Kabupaten Ponorogo Berbasis Web | SI | SI |
| 4. | Perancangan Aplikasi Survive In Disaster Sebagai Media Informasi Tanggap Bencana Berbasis Android | MVK | SI/RPL |
| 5. | Perancangan Dan Penerapan Aplikasi Kasir Dengan Menggunakan Bahasa Pemrograman Php Dan Mysql Di Warung Kopi "Gojeg" | RPL | SI/RPL |
| 6. | Pembuatan Sistem Informasi Laboratorium Teknik Informatika Universitas Ahmad Dahlan | SI | SI |
| 7. | Perancangan Sistem Informasi Akademik Pasraman Berbasis Website | SI | SI |
| 8. | Sistem Pelaporan Dan Monitoring Data Bidang P2P Dinas Kesehatan Kabupaten Sragen | - | SI |
| 9. | Sistem Pengolahan Data Gaji Pegawai CV. Ariesta | - | SI |

| | | | |
|-----|---|-----|--------|
| 48. | Rancang Bangun Sistem Informasi Klinik Kesehatan Berbasis Web | SI | IM/SI |
| 52. | Pengembangan Perangkat Lunak Sistem Informasi Manajemen Skripsi di Jurusan Teknik Elektro | RPL | SI |
| 53. | <i>Pengembangan Aplikasi Manajemen Tugas Skripsi (Studi Kasus: Program Studi Teknik Informatika Universitas Trunojoyo Madura)</i> | RPL | SI |
| 54. | Pembuatan Perangkat Lunak Sistem Informasi Persewaan Barang pada PD. Kharizma Baru | RPL | SI |
| 55. | Perancangan Data Warehouse pada Basis Data Nilai Akademik STT Telkom Menggunakan Model Data Star Schema | SI | SI |
| 60. | Pengembangan Aplikasi untuk Pengelolaan Pengetahuan Dokumen Pribadi Berbasis Ontologi | RPL | SI/KSC |
| 57. | Pembangunan Aplikasi Perhitungan Beban Tugas Dosen Berbasis Web | RPL | SI/RPL |
| 25. | <i>Pengembangan Laboratorium Keamanan Jaringan Berbasis Cloud Computing Menggunakan Container</i> | JKK | JKK |
| 11. | <i>Rancang Bangun Iot Smart Fish Farm Dengan Kendali Raspberry Pi Dan Webcam</i> | - | JKK |
| 12. | Desain dan Implementasi Manajemen Jaringan Berbasis Web untuk Jaringan IP (Internet Protocol) | JKK | JKK |

| | | | |
|-----|--|-----|-----|
| 13. | Peningkatan Keamanan Jaringan dengan Instrusion Detection System Berbasis Distributed System | JKK | JKK |
| 14. | Analisa Multipoint Relaying pada Optimized Link State Routing Protocol untuk Mobile Ad Hoc Network | JKK | JKK |
| 15. | Analisa Keamanan Sistem Komputer terhadap Gangguan DCOM RPC Overflow | JKK | JKK |
| 16. | Aplikasi Berbasis Agent untuk Deteksi Perangkat Keras Komputer melalui Jaringan Lokal | JKK | JKK |
| 17. | Analisa Kualitas Layanan pada Jaringan Internet melalui Mekanisme Protokol Diffserv dengan Algoritma CBQ (Class Based Queuing) | JKK | JKK |
| 18. | Analisa Keamanan Jaringan Wireless LAN dengan Pengujian pada Beberapa Vendor Penyedia Perangkat Jaringan Wireless LAN | JKK | JKK |
| 19. | Implementasi Honeypot sebagai Alat Bantu Deteksi pada Keamanan Jaringan | JKK | JKK |
| 20. | Implementasi Digital Signature pada Sistem Keamanan Mobile Banking | JKK | JKK |
| 21. | <i>Analisa Sistem Klasifikasi Judul Skripsi Menggunakan Metode Naïve Bayes Classifier</i> | KSC | KSC |
| 22. | <i>Prototipe Text Mining Pada Aplikasi Chatting Menggunakan Algoritma Challenge Response</i> | - | KSC |

| | | | |
|-----|--|-----|-----|
| 23. | Klasifikasi Topik Berita Dan Analisis Sentimen Pada Tweets Divisi Humas Polri Dengan Metode Naïve Bayes Classifier | KSC | KSC |
| 24. | Implementasi Metode Latent Dirichlet Allocation Untuk Menentukan Topik Teks Berita | KSC | KSC |
| 10. | Implementasi Information Retrieval System Menggunakan Metode Latent Semantic Indexing Pada Aplikasi Pencarian E-Book | SI | KSC |
| 26. | <i>Sistem Pendukung Keputusan Penentuan Dosen Pembimbing Skripsi Menggunakan Metode Profile Matching (Studi Kasus: STMIK Pelita Nusantara Medan)</i> | KSC | KSC |
| 27. | Integrasi Metode Naïve Bayes dengan Metode Smart Heuristic untuk Klasifikasi Spam | KSC | KSC |
| 28. | Sistem Pendukung Keputusan Penentuan Dosen Pembimbing Dengan Metode Logika Fuzzy | KSC | KSC |
| 29. | Pembangunan Shell Sistem Pakar Klasifikasi dengan Representasi Decision Tree | SI | KSC |
| 30. | Pengklasifikasian E-Mail Menggunakan Klasifikasi Naïve Bayes | KSC | KSC |
| 31. | <i>Perancangan Aplikasi Perbandingan Deteksi Tepi Dalam Citra Digital Dengan Metode Edge Detection Linking Dan Sobel</i> | MVK | MVK |

| | | | |
|-----|---|-----|-----|
| 32. | Animasi 3D Pengenalan Sejarah Waruga di Minahasa | - | MVK |
| 33. | <i>Pembangunan Game Edukasi Mengenal Alat Musik Tradisional Jawa Barat Menggunakan Metode Game Development Life Cycle</i> | - | MVK |
| 34. | Pengenalan Teks pada Objek-Objek Wisata di Sulawesi Utara dengan Teknologi Augmented Reality | MVK | MVK |
| 35. | Restorasi Citra dengan Metode Image Inpainting | - | MVK |
| 36. | Pembuatan Prototipe Animasi Tiga Dimensi dengan Menggunakan Nelmmerse Library | MVK | MVK |
| 37. | Deteksi Wajah Manusia pada Citra Berwarna Menggunakan Fuzzy | | MVK |
| 38. | Perancangan dan Implementasi Teknik Watermarking pada Citra Digital Menggunakan Blowfish dan Message Digest 5 | MVK | MVK |
| 39. | Deteksi Wajah Menggunakan Filter Gabor Wavelet dan Algoritma Genetik | KSC | MVK |
| 40. | Implementasi Grafik Tiga Dimensi dalam Pembuatan Animasi Prisma Segitiga | MVK | MVK |
| 41. | <i>Aplikasi Sistem Pakar Diagnosa Penyakit Gigi Metode Forward Chaining Di UPTD Kesehatan Puskesmas Bangil</i> | SI | IM |

| | | | |
|-----|---|-----|-----|
| 42. | Sistem Pakar Deteksi Penyakit Kanker Menggunakan Pendekatan Naïve Bayesian | SI | IM |
| 43. | Sistem Pakar Penyakit Kulit Berbasis SMS Gateway | SI | IM |
| 44. | Sistem Pakar Diagnosa Penyakit Pencernaan Pada Manusia Menggunakan Metode Certainty Factor Berbasis Web | SI | IM |
| 45. | Metode Mobile Application Sebagai Sistem Monitoring Kondisi Pasien Diabetes Millitus Berbasis Android pada Rumah Sakit Mitra Cirebon | IM | IM |
| 46. | Sistem Pakar Diagnosa Penyakit Anak Menggunakan Metode Forward Chaining Berbasis Mobile | SI | IM |
| 47. | Sistem Pakar Deteksi Penyakit Demam Typhoid dan Demam Berdarah Dengan Menggunakan Algoritma Neural Network dan Metode Adaptive Boosting | KSC | IM |
| 50. | Pemilihan Alat Imaging Technique dalam Mendeteksi Kanker Payudara dengan Menggunakan Simple Additive Weighting Method | IM | IM |
| 49. | Perangkat Lunak Pembantu Penggunaan Komputer Bagi Tuna Netra | RPL | RPL |
| 51. | Rekayasa Perangkat Lunak Peta Wisata Kabupaten Kendal Berbasis Web | RPL | RPL |

| | | | |
|-----|---|-----|-----|
| 56. | Perancangan Perangkat Lunak Tender untuk Jasa Konsultan | RPL | RPL |
| 58. | Studi Metode Terstruktur dalam Rekayasa Perangkat Lunak | RPL | RPL |
| 59. | Rekayasa Ulang Proses Bisnis Sistem Administrasi dan Perpustakaan Sekolah | RPL | RPL |

